NETWORK COMPRESSION VIA NETWORK MEMORY: FUNDAMENTAL PERFORMANCE LIMITS

A Dissertation
Presented to
The Academic Faculty

by

Ahmad Beirami

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2014

NETWORK COMPRESSION VIA NETWORK MEMORY: FUNDAMENTAL PERFORMANCE LIMITS

Approved by:

Professor Faramarz Fekri, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor John Barry
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Matthieu Bloch
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Steven W. McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Howard Weiss
School of Mathematics
Georgia Institute of Technology

Date Approved: 31 March 2014

To my family,

the people whose endless love and support made this work possible.

ACKNOWLEDGEMENTS

Foremost, I would like to thank my research advisor Prof. Faramarz Fekri for his continuous encouragement, advice, and guidance. We started this journey more than five years ago when Dr. Fekri shared with me the basic idea of what today forms the basis of this Ph.D. dissertation. It took a lot of effort and dedication for us to reach where we stand today. Dr. Fekri has kept company as an extremely understanding, kind-hearted, and supportive research advisor along this journey.

I could not have been any luckier to have such wonderful committee members on my Ph.D. dissertation, Professors John Barry, Matthieu Bloch, Steven W. McLaughlin, Raghupathy Sivakumar, and Howard Weiss (in alphabetical order). I have received invaluable feedback on my research from all of them that has greatly improved the quality of my work and its presentation. They have also been extremely supportive of my future research career plans. I got to know Dr. Barry through the coding theory class I took in 2009. Since then I have been always inspired by his teaching style and sharp thinking and have learnt a great deal from him in the occasional discussions that we have had. I have known Dr. Bloch since last year when he moved to Atlanta. During our occasional chats, I have found him very kind and also supportive. Dr. McLaughlin has been involved with my research for the past several years. His excellent insights on the connections between the theory and practice as well as the presentation of my work has resulted in great improvement of my this work to this date. The practical motivation behind the theoretical developments in my Ph.D. dissertation were formed by ideas from Dr. Sivakumar. Ever since we started this project, he has given us excellent feedback in terms of potential applications of the work. Finally, I have known Dr. Weiss since 2009 when I took a class on probability

theory with him. I have always enjoyed how he relates deep mathematical theories to real-world practical applications.

I would like to thank many of the past and present members of center for signal and information processing (CSIP) at Georgia Tech for making a friendly and scientific work place. In particular, I would like to thank my research collaborator Mohsen Sardari. Our stimulating discussions on a daily basis played a major role in identifying the direction of our research. I also enjoyed his company and support outside of the lab whenever I needed it. I thank the fifth floor staff, specially Pat Dixon for her nonstop care during the past five years, and also Jennifer Lunsford and Raquel Plaskett for their support. My thanks also go to Afshin Abdi, Amirhossein Afsharinejad, Erman Ayday, Arash Einolghozati, Liling Huang, Zhinoos Razavi, Nima Torabkhani, and Jun Zou in my research group.

My sincere thanks go to Dr. Saatchi's, Arash & Sara, Banafsheh & Babak, and Laleh & Sasan, as well as my incredible friends Amir H. Atabaki, Bahareh Eidizadeh, Farshid Ghassemi, Reza Pourabolghasem, Mehrsa Raeiszadeh, Venus Saatchi, and Behnaz Zaribaf who have provided me with nonstop support and caring during the past five years in Atlanta. I am also grateful to Alireza Aghasi, Pooyan Amini, Sadegh Bolouki, Sina Haghani, Mohammadreza Imani, Farid Karimpour, Arman Landi, Hessam Mahdavifar, Ali Makhdoumi, Iman Mallakpour, Soroush Mirtalaei, Meysam Moallem, Ali Moayedi, Hamid Nejati, Meisam Razaviyayn, Alireza Saatchi, Amin Sayedi, Behrouz Touri, and many other friends for the enthusiasm and joy they have brought to my life.

Last but not least, I would not be standing here today if it was not for the endless love, support, and motivation from my family throughout my life: my dad Karim, my mom Farah, my sister Elnaz, and my late grandparents Seyed Hossein and Pari. I also would like to thank my aunts and uncles Fariba & Saeed, Afsaneh & Mehdi, and Hossein & Forouzandeh for their support, inspirations, and care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The amount of information that is churned out daily around the world is staggering, and hence, future technological advancements are contingent upon development of scalable acquisition, inference, and communication mechanisms for this massive data. This Ph.D. dissertation draws upon mathematical tools from information theory and statistics to understand the fundamental performance limits of universal compression of this massive data at the packet level using *universal compression* just above layer 3 of the network when the intermediate network nodes are enabled with the capability of memorizing the previous traffic. Universality of compression imposes an inevitable *redundancy* (overhead) to the compression performance of universal codes, which is due to the learning of the unknown source statistics. In this work, the previous asymptotic results about the redundancy of universal compression are generalized to consider the performance of universal compression at the finite-length regime (that is applicable to small network packets). It is proved that the universal compression of small network packets entails significant redundancy up to a factor three more than what would have been asymptotically achieved if the packets were several megabytes in size. To overcome the limits of universal compression for relatively small network packets whenever the network nodes (i.e., the encoder and the decoder) are equipped with memory, *network compression via memory* is proposed as a compression-based solution when massive amounts of previous communication is available to the encoder and the decoder. In a nutshell, network compression via memory learns the patterns and statistics of the payloads of the packets and uses it for compression and reduction of the traffic. Network compression via memory,

with the cost of increasing the computational overhead in the network nodes, significantly reduces the transmission cost in the network. This leads to huge performance improvement as the cost of transmitting one bit is by far greater than the cost of processing it.

CHAPTER I

INTRODUCTION

In several studies, the presence of considerable amounts of correlation in network traffic data is inferred [4, 5, 6, 65, 80, 91, 92]. Specifically, correlation may be broadly defined in three dimensions as (1) correlation within a content being delivered from a source/server to a client, (2) temporal correlation/dependency among different contents from the same source/server across different times being delivered to the same or different clients, and (3) spatial correlation between the content being delivered to the same client from different but correlated sources/servers in the space.

Network traffic abounds with the first dimension of correlation (i.e., correlation within a single content itself). For example, if a traffic contains mostly English text, there is significant correlation in the content. In Chapter 2, the existence of correlation within content for real network data via an experiment is confirmed. The second dimension of correlation (i.e., temporal correlation/dependency among different contents from the same source/server) is present in principle because of the correlation that exists among different contents from the same source/server. For example, suppose a client downloads a content from a web server at time instant t_0. It is quite expected that if the same or another client downloads another content at some other time instant t_1, it would present some correlations with the content at time t_0. Likewise, the third dimension of correlation (i.e., spatial correlation/dependency in different contents from different servers) exists because of the cross-correlation among different contents from different sources/servers. The existence of such correlation has also been confirmed using real traces of network traffic studied by [4, 5, 6, 65, 80, 91, 92]. The third dimension of correlation is present because of the wide variety of applications that involve acquiring data from distributed (i.e., spatially separated) sources that cannot communicate with each other, such as, acquiring digital/analog data from sensors [79, 56, 71, 54, 35], the CEO problem [20, 55], delivery of network packets in a content-centric network [44, 40], acquiring data from femtocell wireless networks [22, 23], and acquiring data chunks from the cloud storage [7, 36].

The large amount of correlation in the network has motivated the development of novel correlation elimination techniques for network traffic data. The present correlation elimination techniques are mostly based on (content) caching mechanisms used by solutions such as web-caching [39], CDNs [58], and P2P applications [57], which target the removal of the first two dimensions of correlation in the network. However, several experiments confirm that the caching approaches, which take place at the application layer, do not efficiently leverage the network correlation which exists mostly at the packet level [92, 91, 65, 4, 6, 5]. To address these issues, ad-hoc methods such as packet-level correlation elimination in which redundant transmissions of segments of a packet that are seen in previously sent packets are avoided have been considered in a few recent studies [6, 5]. However, these techniques are limited in scope and can only eliminate exact duplications from larger segments of the packets while ignoring the correlation due to the statistical dependency between the symbols inside the packet.

Please note that universal compression schemes may also be considered as potential *end-to-end* correlation elimination techniques for network traffic data.[1] Since Shannon's seminal work on the analysis of communication systems, many researchers have contributed toward the development of compression schemes with the average codeword length as close as possible to the entropy, i.e., the fundamental compression

[1] Please note that in end-to-end universal compression the encoding of the packet is performed at the server and the decoding of the packet is performed at the client without using the intermediate nodes in the network, and hence, the name end-to-end.

limit [28, 94, 95, 88, 87, 31, 10, 72]. Provided that the statistics of the information source are *known*, Huffman block coding achieves the entropy of a sequence with a negligible redundancy smaller than 1 bit, which is due to the integer length constraint on the codewords [81, 84]. In the network traffic, however, a priori knowledge on the statistics of the source cannot be assumed while it is still desired to compress the *unknown* stationary ergodic source to its entropy rate. This is known as the *universal* compression problem [28, 95, 87, 88, 94, 32, 59, 61, 60, 51, 13, 26, 47, 41, 8, 86, 52, 16, 59].

Universality of compression imposes an inevitable *redundancy*, which is due to the learning of source statistics. This redundancy depends on the richness of the class of the sources with respect to which the code is universal [86, 52, 16, 59]. In [77], Shields showed that a universal redundancy for the class of stationary ergodic sources does not exist by constructing a stationary ergodic source whose redundancy rate dominates any given rate. Therefore, in this work, the study is focused on the fairly general class of parametric sources for which universal redundancy rates are known to exist [59, 60]. The asymptotic behavior of the average redundancy of prefix-free codes on the class of parametric sources has been investigated in the past and the main terms of the redundancy have been exactly characterized [86, 60, 13]. In particular, Merhav and Feder also derived a probablistic lower bound on the average redundancy resulting from the compression of a sequence of length n from the family of the parametric sources, where the source parameter follows the capacity achieving prior (i.e., Jeffreys' prior) [52].

In Chapter 2, a tight probabilistic converse on the redundancy of the compression of a finite-length sequence from the family of parametric sources is presented. Achievability of the the derived bound is also provided, which leads to the exact characterization of the average redundancy of the prefix-free codes. As shall be discussed in Chapter 2, our results collectively demonstrate that the compression of finite-length sequences (up to hundreds of kilobytes) incurs a significant redundancy (i.e., overhead). This places a strict performance limit for many practical applications including network packet compression. In other words, universal compression techniques require very long data before they can effectively remove correlation from the network packets. Further, the end-to-end traditional information theoretic techniques would only attempt to deal with the first type of correlation mentioned above (i.e., correlation within a content) and lacks the structure to leverage the second dimension of correlation.

In Chapter 3, *network compression via network memory* is proposed as a promising solution to overcome the limitations of universal compression of small network packets. The basic premise of the network compression relies on the rational that network elements, such as routers or other intermediate nodes, can be enabled to memorize the traffic and learn about network source contents as they forward the packets. This knowledge about source contents is then used by a properly designed memory-assisted compression mechanism to represent new contents more efficiently (i.e., smaller codewords). In a nutshell, equipping the network with network memory elements will enable memorization of the source traffic as it flows naturally (or by design) through the network. Hence, memory enabled nodes can learn the source data statistics which can then be used (as side information) toward reducing the cost of describing the source data statistics in the universal compression. The idea is based on the fact that training data can be useful in decreasing the redundancy in universal compression (cf. [47, 45, 93] and the references therein). In this work, the packet memorization gain problem is theoretically formulate and the fundamental limits of network compression via network memory is investigated.

Chapter 4 targets the third dimension of the correlation in the network, i.e., the correlation between the content being delivered to the same client from spatially separated but correlated sources/servers. Although there are several formulations for the multi-terminal compression problem in the literature (cf. [79, 56, 54, 71, 35, 20, 55] and the references therein), there are several emerging scenarios (e.g., the content-centric networks and wireless femtocell networks) that do not fall into the realm of the existing multi-terminal source coding problems (i.e., Slepian-Wolf, Wyner-Ziv, CEO problem, etc). Previous work is mostly concerned about the compression of sequences that bear symbol-by-symbol correlation. On the other hand, the focus of this work is on the universal compression of the data traffic from multiple sources with correlated parameter vectors which is a different notion of correlated sources. In Chapter 4, *universal compression of distributed parametric sources with correlated parameter vectors* is introduced and studied. As the most basic case, two parametric sources S_1 and S_2 with unknown but correlated parameter vectors are assumed to communicate with a destination node M. Note that y^m and x^n are generated as *independent* samples of S_1 and S_2 (given the source parameter vectors are known), respectively. However, when the source parameter vectors are unknown, y^m and x^n are *correlated* with each other through the information they contain about the unknown but correlated source parameter vectors. We wish to leverage this correlation in the encoder of S_2 and the decoder of M in the decoding of x^n by using the side information sequence y^m (from S_1) in order to reduce the average codeword length of x^n. This problem can also be viewed as universal compression with training data (that is taken from a source with correlated source parameter), where the training data is only available to the decoder. This problem can also be viewed as universal compression [28, 94, 88], with decoder side only training data.

In Chapter 5, the source model is extended to a more realistic model for the compression of network packets. Thus far, in the characterization of the fundamental benefits of memory-assisted universal compression, it was assumed that the traffic was generated by a single stationary ergodic source. Clearly, a single stationary ergodic source does not fully model a real content generator server (for example the CNN news website in the Internet). Instead, a better model is to view every content generator server as mixture of several information sources whose true statistical models are not readily available. This issue has been considered in [70] by Sardari *et. al.*, where a memorization and clustering technique for compression is proposed. However, several questions still remained open that are addressed in the fourth chapter of this work. It is proved that the benefits of the *optimal* memory-assisted universal compression are obtained by the clustering of the side information sequences in the memory. The fundamental limits of the performance improvement that is expected from the joint memorization and clustering versus the memorization without clustering are derived. Hints are provided on how clustering scheme must be realized to achieve the optimal compression performance with the joint memorization and clustering. Finally, simulation results on man-made data as well as real traffic traces prove the usefulness of clustering for memory-assisted compression.

In Chapter 6, the prefix constraint on the code is dropped. Thus far, the focus of our work has been on prefix-free codes (uniquely decodable codes) where the length function l_n is required to satisfy Kraft's inequality. Kraft's inequality ensures that when several blocks of length n are encoded using c_n there exists a uniquely decodable code with length function l_n such that all the blocks can be uniquely decoded. For the prefix-free codes, it is straightforward to demonstrate that the optimal average codeword length is bounded below by the entropy. On the other hand, there are several applications that do not require the unique decodability of the concatenated blocks since the beginning and the end of each block is already known. For example, in the compression of network packets, the end of each IP packet is already determined by the header of the packet. Therefore, in these cases, the unique decodability condition is indeed too restrictive. Instead, if the mapping c_n is injective, it is ensured that one block of length n can be uniquely decoded. These codes are known as one-to-one codes. It has been recently shown that the average codeword length of one-to-one codes can be significantly below the entropy, as exactly characterized by Szpankowski and Verdu [83]. *Universal* one-to-one codes (without prefix constraints) are investigated to obtain the optimal universal coding strategy as well as the fundamental limits (i.e., the characterization of the achievable rate regions and the cost of universality). Further, the *network compression via network memory* setup is extended to the codes without prefix constraint and the improvements obtained from memorization of previous packets on universal one-to-one codes are characterized.

Finally, it is worth noting the scope of the network-compression benefits raised in this work is significant since file sharing and web data is predicted to comprise more than 45% of network traffic by year 2016 [1] for which, the packet correlation may reach as high as 40% [92, 5, 75] which cannot be modelled/exploited using the conventional distributed source coding approaches. Further, the memory-assisted universal compression can also be applied to a range of applications such as storage reduction of cloud and distributed storage systems, traffic reduction for Internet Service Providers, and power and bandwidth reduction of wireless communications networks (e.g., wireless sensors networks, cellular mobile networks, hot spots). We will only focus on its application to the compression of packets from a single source in this work. A summary of the achievements and directions for future research are presented in Chapter 7.

CHAPTER II

UNIVERSAL COMPRESSION IN FINITE LENGTH

Ever since entropy rate was shown to be the lower bound on the average compression rate of any stationary source using prefix-free codes, many researchers have contributed toward the development of prefix-free codes with average codeword length approaching the entropy of the sequence. Provided that the statistics of the information source are *known*, Huffman block coding achieves the entropy of a sequence with a negligible redundancy smaller than 1 bit, which is due to the integer length constraint on the codewords [81, 84]. In many applications, however, the sequences to be compressed does not follow a fixed statistical model. Thus, the underlying probability distribution of the sequence is a priori unknown requiring the compression to be universal [28, 94, 88, 63, 32, 31, 10, 14, 47].

In [77], Shields showed that a universal redundancy rate for the class of stationary ergodic sources does not exist by constructing a stationary ergodic source whose redundancy rate dominates any given rate. Therefore, in this work, our study is focused on the fairly general class of parametric sources for which universal redundancy rates are known to exist and are asymptotically characterized [59, 60]. The asymptotic behavior of the average redundancy of prefix-free codes on the class of parametric sources has been extensively studied (cf. [86, 60, 13, 52, 24] and the references therein) and the main term of the average redundancy has been exactly characterized to be $d/2 \log n + O(1)$ [86, 60, 13]. In particular, Merhav and Feder in [52] also derived a probablistic lower bound on the average redundancy resulting from the compression of a sequence of length n from the family of the parametric sources, where the source parameter follows the capacity achieving prior (i.e., Jeffreys' prior in the case of parametric sources).

In this chapter, we provide a tight converse bound on the average redundancy of two-part codes and by providing the achievability of the bound, we exactly characterize the average redundancy of the optimal two-part code for the family of the parametric sources. In a two-part code, the coding is performed in two stages. The first part of the code provides an estimation of the parameter vector obtained from the sequence to be compressed. The second part of the code is the best (non-universal) codeword for the sequence based on the estimated parameter in the first part of the code. We consider both ordinary and normalized two-part codes. The ordinary two-part codes are not optimal as they asymptotically incur an extra redundancy term (which is characterized in this chapter) on top of the average minimax redundancy of the prefix-free codes. On the other hand, the normalized two-part codes are optimal in the sense that they achieve the average minimax redundancy of the prefix-free codes. All of the known universal codes that achieve the average minimax redundancy, such as the context tree weighting algorithm, can be cast as normalized two-part codes, and hence, fall within the realm of our results. Additionally, we conjecture that our results also hold for all universal prefix-free codes for parametric sources that satisfy Kraft's inequality.

Our contributions in this chapter are summarized in the following:

- A tight probabilistic converse (lower bound) on the average redundancy of the compression of a finite-length sequence from the family of parametric sources for the class of two-part codes is derived when the unknown source parameter follows the least favorable prior.

- Achievability of the the derived bound is provided, which leads to the exact probabilistic characterization of the average redundancy of the two-part codes under the least favorable prior.

- The family of ordinary two-part codes are studied and their performance loss in comparison with the normalized two-part codes is exactly characterized.

The rest of this chapter is organized as follows. The background review and the related work is provided in Section 2.1. In Section 2.2, the formal statement of the problem of redundancy for finite-length universal compression of parametric sources using the two-part codes is presented. In Section 2.3, our main results on the average redundancy for universal compression of finite-length sequences are provided. In Section 2.4, the significance of our results are demonstrated through several examples using memoryless sources as well as finite memory Markov sources. Finally, the conclusion is given in Section 2.5.

2.1 Background Review

In the following, we describe our source model together with necessary notations and related work. Denote $\mathcal{A}$ as a finite alphabet. Let the parametric source be defined using a d-dimensional parameter vector $\theta = (\theta_1, ..., \theta_d)$, where d denotes the number of the source parameters. Denote μ_θ as the probability measure defined by the parameter vector θ on sequences of length n. We also use the notation μ_θ to refer to the parametric source itself. We assume that the d parameters are unknown and lie in the d-dimensional space $\Lambda \subset \mathbb{R}^d$. Denote $\mathcal{P}_\Lambda^d$ as the *family* of parametric sources with d-dimensional unknown parameter vector θ such that $\theta \in \Lambda$. The family $\mathcal{P}_\Lambda^d$ contains all source models that have a *minimal* representation with a d-dimensional parameter vector θ. We use the notation $x^n = (x_1, ..., x_n) \in \mathcal{A}^d$ to represent a sequence of length n (which is assumed to be a realization of the random vector X^n that follows μ_θ unless otherwise stated). Let $H_n(\theta)$ be the source entropy given parameter vector θ, i.e.,

$$H_n(\theta) \triangleq \mathbf{E} \log\left(\frac{1}{\mu_\theta(X^n)}\right) = \sum_{x^n} \mu_\theta(x^n) \log\left(\frac{1}{\mu_\theta(x^n)}\right).^{1} \tag{1}$$

In this dissertation $\log(\cdot)$ always denotes the logarithm in base 2.

In this work, we consider the family of block codes that map any n-vector to a variable-length binary sequence [81]. Next, we present the notions of strictly lossless and almost lossless source codes, which will be needed in the sequel.

Definition 2.1.1. *The code $c_n : \mathcal{A}^n \to \{0,1\}^*$ is called strictly lossless (also called zero-error) if there exists a reverse mapping $d_n : \{0,1\}^* \to \mathcal{A}^n$ such that*

$$\forall x^n \in \mathcal{A}^n : \quad d_n(c_n(x^n)) = x^n.$$

Most of the practical data compression schemes are examples of strictly lossless codes, namely, the arithmetic coding [49], Huffman [38], Lempel-Ziv [94, 95], and Context-Tree-Weighting algorithm [88].

On the other hand, in the later chapters of the work, we are also concerned with almost lossless source coding, which is a the slightly weaker notion of the lossless case.

Definition 2.1.2. *The code $c_n^{p_e} : \mathcal{A}^n \to \{0,1\}^*$ is called almost lossless with permissible error probability $p_e(n) = o(1)$, if there exists a reverse mapping $d_n^{p_e} : \{0,1\}^* \to \mathcal{A}^n$ such that*

$$\mathbf{E}\{\mathbf{1}_e(X^n)\} \leq p_e(n),$$

where $\mathbf{1}_e(x^n)$ denotes the error indicator function, i.e,

$$\mathbf{1}_e(x^n) \triangleq \begin{cases} 1 & d_n^{p_e}(c_n^{p_e}(x^n)) \neq x^n, \\ 0 & \textit{otherwise.} \end{cases}$$

[1] Throughout this chapter all expectations are taken with respect to the distribution μ_θ induced by the true unknown parameter vector θ.

The almost lossless codes allow a non-zero error probability $p_e(n)$ for any finite n while they are *almost surely* asymptotically error free. Note that almost lossless codes with $p_e(n) = 0$ are indeed strictly lossless codes. Thus, we also use the notation c_n^0 to denote a strictly lossless code. The proofs of Shannon [74] for the existence of entropy achieving source codes are based on almost lossless random codes. The proof of the Slepian-Wolf theorem [79] also uses almost lossless codes. Further, all of the practical implementations of SW source coding are based on almost lossless codes (cf. [56, 71]). We stress that the almost lossless source coding is distinct from the lossy source coding (i.e., the rate-distortion theory). In the rate-distortion theory, a code is designed to achieve a given distortion level asymptotically as the length of the sequence grows to infinity. Therefore, since the almost lossless coding asymptotically achieves a zero-distortion, in fact, it coincides with the special case of zero-distortion in the rate-distortion curve.

Denote $l_n(x^n) = l(c_n, x^n)$ as the length function that describes the length of the codeword associated with the sequence x^n. In most of the work except Chapter 6, we only consider prefix-free length functions, which in turn satisfy Kraft's inequality, i.e.,

$$\sum_{x^n \in \mathcal{A}^n} 2^{-l_n(x^n)} \leq 1 \tag{2}$$

Please note that we ignore the integer constraint on the length function, which results in a negligible redundancy upper bounded by 1 bit analyzed exactly in [30, 81]. Denote L_n as the set of all prefix-free length functions on an input sequence of length n that satisfy Kraft's inequality.

Let $\mathcal{I}_n(\theta)$ be the Fisher information matrix for parameter vector θ,

$$\mathcal{I}_n(\theta) \triangleq \left\{\mathcal{I}_n^{ij}(\theta)\right\} = \frac{1}{n \log e}\mathbf{E}\left\{\frac{\partial^2}{\partial\theta_i \partial\theta_j} \log\left(\frac{1}{\mu_\theta(X^n)}\right)\right\}. \tag{3}$$

Fisher information matrix quantifies the amount of information, on the average, that each symbol in a sample sequence from the source conveys about the unknown source parameters. We assume that the following conditions hold:

1. Λ is a compact convex subspace of $\mathbb{R}^d$.

2. The parametric family $\Lambda \subset \mathbb{R}^d$ has a minimal d-dimensional representation.

3. All elements of the Fisher information matrix $\mathcal{I}_n(\theta)$ are continuous in Λ.

4. $\lim_{n\to\infty} \mathcal{I}_n(\theta)$ exists and the limit is denoted by $\mathcal{I}(\theta)$.

5. $\int_{\theta\in\Lambda} |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta < \infty$.

Let $r_n(l_n, \theta, x^n)$ denote the redundancy of the code with length function l_n for the source parameter vector θ on the individual sequence x^n, which is defined as

$$r_n(l_n, \theta, x^n) \triangleq l_n(x^n) - \log\left(\frac{1}{\mu_\theta(x^n)}\right), \tag{4}$$

where $\log(1/\mu_\theta(x^n))$ is the length of the corresponding optimal non-universal code, which in turn minimizes the average codeword length of prefix-free codes. Note that the redundancy for an individual sequence x^n need not be necessarily non-negative. Previous works [28, 78] have studied the worst-case minimax redundancy defined as

$$\bar{r}_n \triangleq \inf_{l_n\in L_n} \sup_{\theta\in\Lambda} \max_{x^n\in\mathcal{A}^n} \left\{r_n(l_n, \theta, x^n)\right\}. \tag{5}$$

The worst-case minimax redundancy characterizes the performance of the compression on the worst-case individual sequence [78, 64, 86, 13, 33, 41, 85, 30]. It has been shown that the leading term in r_n is asymptotically $\frac{d}{2}\log n$. In particular, Szpankowski derived the asymptotic behavior of the worst-case minimax redundancy and precisely derived all the terms up to $O(n^{-3/2})$ [30]. The worst-case minimax redundancy, by definition, provides a good performance measure whenever bad compression performance is not tolerated on any individual sequence. However, in most applications, we are interested in reducing the average number of the transmitted bits, and hence, the average redundancy is a better performance metric as opposed to worst-case redundancy.

Denote $R_n(l_n, \theta)$ as the average redundancy of the code on a sequence of length n, defined as the difference between the expected (average) codeword length and the entropy. That is

$$R_n(l_n, \theta) \triangleq \mathbf{E} r_n(l_n, \theta, X^n) = \mathbf{E} l_n(X^n) - H_n(\theta). \tag{6}$$

The average redundancy is clearly non-negative, where the zero-redundancy is achieved by the optimal non-universal code. Further, a code is called universal if its average codeword length normalized to the sequence length uniformly converges to the source entropy rate, i.e., $\lim_{n\to\infty} \frac{1}{n} R_n(l_n, \theta) = 0$ for all $\theta \in \Lambda$.

Rissanen proved an asymptotic lower bound on the universal compression of information sources with d parameters as [60, 61]:

Theorem 2.1.3. [60]: *In the universal compression of the family $\mathcal{P}^d_\Lambda$, for all parameter vectors $\theta \in \Lambda$, except in a set of asymptotically Lebesgue volume zero, and for any $\epsilon > 0$ we have*

$$\lim_{n\to\infty} \frac{R_n(l_n, \theta)}{\frac{d}{2}\log n} \geq 1 - \epsilon. \tag{7}$$

Rissanen's bound is asymptotically tight up to $o(\log n)$ as it is shown to be achievable as well [60, 88]. Similar results were later derived for more general classes of sources [32, 52]. While Theorem 2.1.3 determines the asymptotic fundamental limits of the universal compression of parametric sources, it does not provide much insight for the case of short length sequences. Moreover, the result excludes an asymptotically volume zero set of parameter vectors θ that has non-zero volume for any finite n that needs to be characterized before anything can be stated regarding the coding performance in the short-length regime.

Let the average minimax redundancy for a code with length function l_n be defined as [24, 90, 29]

$$\bar{R}_n \triangleq \inf_{l_n \in L_n} \sup_{\theta \in \Lambda} R_n(l_n, \theta). \tag{8}$$

The average minimax redundancy is concerned with the performance of the best code for the worst source parameter chosen by the nature. In [24], Clarke and Barron derived the average minimax redundancy $\bar{R}_n$ for memoryless sources, later generalized in [8] by Atteson for Markov sources, as the following:

Theorem 2.1.4. [24, 8]: *The average minimax redundancy is asymptotically given by*

$$\bar{R}_n = \frac{d}{2}\log\left(\frac{n}{2\pi e}\right) + \log\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta + O\left(\frac{1}{n}\right). \tag{9}$$

The average minimax redundancy characterizes the maximum redundancy over the space Λ of the parameter vectors. However, it does not say much about the rest of the space of the parameter vectors. Gallager showed that if $\mu_\theta(x^n)$ is a measurable function of θ for all x^n, the average minimax redundancy is equal to the average maximin redundancy as well as the capacity of the channel between the parameter vector θ and the sequence x^n, i.e., $\bar{R}_n = \max_\omega I(\theta; X^n)$, where $\omega(\cdot)$ is a probability measure on the space Λ where the parameter vector θ resides [34, 28, 52]. Please note that average maximin redundancy is defined as the following.

$$\underline{R}_n = \sup_p \inf_{l_n \in L_n} \int_{\theta\in\Lambda} R_n(l_n, \theta) p(\theta) d\theta \tag{10}$$

The average maximin redundancy is associated with the best code under the worst prior on the space of parameter vectors (i.e., the capacity achieving Jeffreys' prior).

The average minimax redundancy is achieved by a code that assumes that the parameter vector θ follows the capacity achieving prior, i.e., Jeffreys' prior is both capacity achieving and minimax optimal [52]. Jeffreys' prior is given by [90]

$$p_J(\theta) \triangleq \frac{|\mathcal{I}(\theta)|^{\frac{1}{2}}}{\int_{\lambda\in\Lambda} |\mathcal{I}(\lambda)|^{\frac{1}{2}} d\lambda}. \tag{11}$$

Rissanen further proved that the redundancy for individual sequences defined in (4), except for a Lebesgue volume zero set of sequences, is asymptotically given as $\frac{d}{2}\log n +$

$o(\log n)$. In [52], Merhav and Feder extended Rissanen's result to more general classes of sources and demonstrated that asymptotically almost all sources chosen using the capacity achieving prior (Jeffrey's prior in the case of parametric sources) have a redundancy no smaller than the average minimax redundancy of prefix-free codes. In particular, Merhav and Feder's result directly implies the following finite-length result about the average redundancy of prefix-free codes on the parametric sources.

Theorem 2.1.5. [52]: *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family $\mathcal{P}_\Lambda^d$ of parametric sources. Then, $\forall \epsilon > 0$, we have*

$$\mathbf{P}_\theta \left[\frac{R_n(l_n, \theta)}{\bar{R}_n} \leq 1 - \epsilon \right] \leq e 2^{-\epsilon \bar{R}_n}. \tag{12}$$

Theorem 2.1.5 provides with a strong lower bound on the average redundancy of prefix-free codes on the parametric sources when the unknown source parameter follows the least favorable Jeffreys' prior for all n. Further, it is not difficult to deduce Theorem 2.1.3 from this result.

2.2 Universal Compression using Two-Part Codes

In this work, we consider the fairly general family of two-part prefix-free codes. We provide a tight probabilistic converse bound on the average redundancy of two-part codes when the unknown source parameter vector follows the least favorable Jeffreys' prior. We further prove the achievability of our bound leading to the characterization of the achievable region for coding using two-part codes. This is an important result as most of the capacity achieving codes, such as the context tree weighting (CTW) [88], are the concatenation of a *predictor* and arithmetic coding (which is a practical sequential version of the optimal code), and hence, are included in the two-part codes. Next, we state the average redundancy problem in the finite-length regime, where we will consider both ordinary two-part and normalized two-part codes.

2.2.1 Ordinary Two-Part Codes

In an ordinary two-part code, to encode the sequence x^n, the compression scheme first estimates the source parameter vector. Then, in the second part, the sequence x^n is encoded using an optimal code for the estimated parameter vector [62, 11, 37]. Let $\Phi^m = \{\phi_1, ..., \phi_m\}$ denote the set of all possible estimates of the unknown source parameter vector θ, where $\phi_i \in \Lambda$ and $m \in \mathbb{N}$ is a positive integer denoting the number of the possible estimate points. Please note that $M = M(n)$ is a function of the sequence length. The two-part code is given by

$$c_n^{2p}(x^n, \phi_i) = [\hat{c}_n(\phi_i);\ \dot{c}_n(x^n, \phi_i)]\,, \tag{13}$$

where $\hat{c}_n(\phi_i)$ denotes the optimal prefix-free code that is used for describing the estimate ϕ_i of the parameter vector θ, and $\dot{c}_n(x^n, \phi_i)$ is the optimal prefix-free non-universal code for x^n given the optimal estimate of the parameter vector is ϕ_i.

Let $\hat{l}_n : \Phi^M \to \mathbb{R}$ denote the prefix-free codeword length function for the estimated parameter vectors $\phi_i \in \Phi^m$. Let $\pi = (\pi(\phi_1), \dots, \pi(\phi_M))$ denote an arbitrary distribution on the set of estimate points in Φ^m. Hence, as discussed earlier, the optimal length function $hatl_n$ is given by $\hat{l}_n(\phi_i) = -\log(\pi(\phi_i))$. On the other hand, the optimal (non-universal) codeword length $\dot{l}_n$ for the sequence x^n, given the parameter vector ϕ_i, is simply given by the optimal non-universal code, i.e., $\dot{l}_n(x^n, \phi_i) = -\log\left(\mu_{\phi_i}(x^n)\right)$. Accordingly, denote l_n^{2p} as the two-part (universal) length function for the compression of sequences of length n, which is defined as

$$l_n^{2p}(x^n) = \log\left(\frac{1}{\pi(\phi^\star)}\right) + \log\left(\frac{1}{\mu_{\phi^\star}(x^n)}\right), \tag{14}$$

where

$$\begin{aligned}\phi^\star = \phi^\star(x^n, \Phi^m) &\triangleq \arg\min_{\phi_i \in \Phi^m} \left\{\log\left(\frac{1}{\pi(\phi_i)\mu_{\phi_i}(x^n)}\right)\right\} \\ &= \arg\max_{\phi_i \in \Phi^m} \left\{\pi(\phi_i)\mu_{\phi_i}(x^n)\right\}. \end{aligned} \tag{15}$$

Let L_n^{2p} be the set of all two-part codes that could be described using (14) for an arbitrary $m \in \mathbb{N}$ and arbitrary set of estimate points Φ^m and an arbitrary probability vector π. Increasing the budget m for the identification of the unknown source parameters results in a growth in the number of estimate points, and hence, smaller $\dot{l}_n(x^n, \phi^\star)$ on the average due to the more accurate estimation of the unknown source parameter vector. On the other hand, m plays a direct role in the first-term of compression overhead, i.e., $-\log(\pi(\phi^\star))$ in (14). Therefore, it is desirable to find the optimal m and π that minimize the total expected codeword length for a large fraction of sources that follow the least favorable Jeffreys' prior, which is

$$\mathbf{E}l_n^{2p}(X^n) = \mathbf{E}\log\left(\frac{1}{\pi(\phi^\star)}\right) + \mathbf{E}\log\left(\frac{1}{\mu_{\phi^\star}(X^n)}\right). \tag{16}$$

Let the average redundancy of two-part codes be defined as $\bar{R}_n(l_n^{2p}) \triangleq \mathbf{E}l_n^{2p}(X^n) - H_n(\theta)$, which can be expressed as

$$R_n(l_n^{2p}, \theta) = \mathbf{E}\log\left(\frac{1}{\pi(\phi^\star)}\right) + \mathbf{E}\log\left(\frac{\mu_\theta(X^n)}{\mu_{\phi^\star}(X^n)}\right). \tag{17}$$

Please note that $\phi^\star = \phi^\star(x^n, \Phi^m)$ is implicitly a function of the sequence x^n, and thus, the calculation of the above expectations is nontrivial. In Section 2.3.1, we characterize the average redundancy of ordinary two-part codes in (17) in the small sequence length regime and derive a probabilistic lower bound on the average redundancy. Further, let $\bar{R}_n^{2p}$ denote the average minimax redundancy of the ordinary two-part codes, i.e.,

$$\bar{R}_n^{2p} = \inf_{l_n^{2p} \in L_n^{2p}} \sup_{\theta \in \Lambda} R_n(l_n^{2p}, \theta). \tag{18}$$

In Section 2.3.2, we precisely characterize $\bar{R}_n^{2p}$ and compare it with the average minimax redundancy of the prefix-free codes in general.

2.2.2 Normalized Two-Part Codes

Thus far, we presented the universal compression using ordinary two-part codes. In an ordinary two-part code, we already have some knowledge about the sequence

x^n through the optimally estimated parameter $\phi^\star(x^n, \Phi^M)$ (i.e., maximum likelihood estimation). This knowledge can be leveraged for encoding x^n using the second part of the code via length function $\dot{l}_n(x^n, \phi^\star)$. In fact, the ordinary two-part length function in (14) defines an incomplete coding length, in the sense that it does not achieve the equality in Kraft's inequality. Hence, it is straightforward to show that it does not achieve the average minimax redundancy of the prefix-free codes either [15, 37]. On the other hand, conditioned on $\phi^\star(x^n, \Phi^M)$, the length of the codeword for x^n may be further decreased if $\hat{\mu}_{\phi^\star}(x^n)$ is used instead of $\mu_{\phi^\star}(x^n)$ [62] so that the coding scheme becomes complete. This is due to the fact that the length function of sequence x^n is now associated with a probability distribution over the sequences of length n in $\mathcal{A}^n$. the This complete version of the this coding system is called a normalized two-part code.

Let $S(\gamma, \Phi^m) \subset \mathcal{A}^n$ denote the set of all $x^n \in \mathcal{A}^n$ for which the optimally estimated parameter is $\gamma \in \Phi^m$, i.e.,

$$S(\gamma, \Phi^m) \triangleq \{x^n \in \mathcal{A}^n : \ \phi^\star(x^n, \Phi^m) = \gamma\}. \tag{19}$$

Further, let $A(\gamma, \Phi^m)$ denote the total probability measure of all sequences in the set $S(\gamma, \Phi^m)$ induced by μ_γ, i.e.,

$$A(\gamma, \Phi^m) = \sum_{x^n \in S(\gamma, \Phi^m)} \mu_\gamma(x^n). \tag{20}$$

Thus, the knowledge of $\phi^\star(x^n, \Phi^M)$ in fact changes the probability distribution of the sequence. Denote $\hat{\mu}_{\phi^\star}(x^n)$ as the probability measure of x^n induced by the parameter vector $\phi^\star$ given that $\phi^\star(x^n, \Phi^M)$ is known, which is given by

$$\hat{\mu}_{\phi^\star}(x^n) = \frac{\mu_{\phi^\star}(x^n)}{A(\phi^\star, \Phi^m)}. \tag{21}$$

Note that $\hat{\mu}_{\phi^\star}(x^n) \geq \mu_{\phi^\star}(x^n)$ due to the fact that $A(\phi^\star, \Phi^m) \leq 1$. Let $\ddot{l}_n(x^n, \phi_i)$ be the codeword length corresponding to the conditional probability distribution, which is given by

$$\ddot{l}_n(x^n, \phi_i) = \log \left(\frac{A(\phi_i, \Phi^m)}{\mu_{\phi_i}(x^n)} \right). \tag{22}$$

Denote l_n^{n2p} as the normalized two-part length function for the compression of sequences of length n using the normalized maximum likelihood. Then, we have

$$l_n^{n2p}(x^n) = \log \left(\frac{A(\phi^\star, \Phi^m)}{\pi(\phi^\star)\mu_{\phi^\star}(x^n)} \right), \tag{23}$$

Furthermore, it is clear from this definition that the normalized two-part codes define a complete coding system in the sense that the universal lengths satisfy Kraft's inequality with the equality sign.

Denote L_n^{n2p} as the set of the normalized two-part codes that are described using (23). Let $\bar{R}_n^{n2p}$ be the average minimax redundancy of the normalized two-part codes, i.e.,

$$\bar{R}_n^{n2p} = \inf_{l_n^{n2p} \in L_n^{n2p}} \sup_{\theta \in \Lambda} R_n(l_n^{n2p}, \theta). \tag{24}$$

Rissanen demonstrated that this normalized version of two-part codes is in fact optimal in the sense that the average minimax redundancy of the normalized two-part codes is equal to that of the general prefix-free codes [64]. In other words, $\bar{R}_n^{n2p} = \bar{R}_n$, where $\bar{R}_n$ is the average minimax redundancy of the prefix-free codes defined in (9). In Section 2.3.3, our goal is to investigate the performance of the normalized two-part codes using (29). In particular, we derive a probabilistic lower bound on the average redundancy for the compression of parametric sources using normalized two-part codes.

2.3 Main Results on the Redundancy

In Section 2.3.1, we present a converse bound on the probability of the event that the average redundancy of ordinary two-part codes is smaller than any fraction of $\frac{d}{2} \log n$ when the unknown source parameter follows the least favorable Jeffreys' prior. In Section 2.3.2, we precisely characterize the average minimax redundancy of ordinary two-part codes. In Section 2.3.3, we generalize the main result on the average redundancy to the normalized two-part codes. In Section 2.3.4, we tailor the main results to the class of finite-alphabet memoryless sources and restate the main results. In Section 2.3.5, we describe a simple construction that achieves the derived converse bounds.

2.3.1 Two-Part Code Redundancy

In this section, we restrict the code to the set of ordinary two-part length functions, i.e., $l_n^{2p} \in L_n^{2p}$. We derive an upper bound on the probability of the event that a sequence of length n from the parametric source μ_θ is compressed with redundancy smaller than $(1-\epsilon)\frac{d}{2}\log n$ for any given n and ϵ. In other words, we find an upper bound on $\mathbf{P}_\theta[R_n(l_n^{2p},\theta) < (1-\epsilon)\frac{d}{2}\log n]$, which in turn sets a probabilistic upper bound on the the average redundancy. Let

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \tag{25}$$

denote Euler's gamma function.

Theorem 2.3.1. *Consider the universal compression of the family of parametric sources $\mathcal{P}_\Lambda^d$ with the unknown parameter vector θ that follows Jeffreys' prior. Let ϵ be a real number. Then,*

$$\mathbf{P}_\theta\left[\frac{R_n(l_n^{2p},\theta)}{\frac{d}{2}\log n} \le 1-\epsilon\right] \le \frac{C_d}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}}d\theta}\left(\frac{d}{en^\epsilon}\right)^{\frac{d}{2}} f(n), \tag{26}$$

where $f(n) = 1 + o(1)$ and C_d is the volume of the d-dimensional unit ball given by

$$C_d = \frac{\Gamma\left(\frac{1}{2}\right)^d}{\Gamma\left(\frac{d}{2}+1\right)}. \tag{27}$$

As we shall see in the following, the proof of Theorem 2.3.1 is constructive, and hence, the lower bound is indeed asymptotically achievable if we ignore the integer constraint on the codeword length.

First, please note that $l_n^{n2p}(x^n) \leq l_n^{2p}(x^n)$. Further, we have

$$l_n^{2p}(x^n) - l_n^{n2p}(x^n) \geq \log \left(\frac{1}{A(\phi^\star, \Phi^m)} \right). \tag{28}$$

Therefore, the average redundancy of the normalized two-part scheme is given by

$$R_n(l_n^{n2p}, \theta) \leq R_n(l_n^{2p}, \theta) - \mathbf{E} \log \left(\frac{1}{A(\phi^\star, \Phi^m)} \right). \tag{29}$$

To prove Theorem 2.3.1, first, we need to bound the average redundancy in (17). To proceed, let ϕ° be defined as

$$\phi^\circ = \phi^\circ(\theta, \Phi^M) \triangleq \arg \min_{\phi_i \in \Phi^m} D_n(\mu_\theta || \mu_{\phi_i}). \tag{30}$$

where $D_n(\mu_\theta || \mu_{\phi^\circ})$ is the non-negative Kullback–Leibler divergence between the probability measures μ_θ and μ_{ϕ° given by

$$D_n(\mu_\theta || \mu_{\phi^\circ}) = \mathbf{E} \log \left(\frac{\mu_\theta(x^n)}{\mu_{\phi^\circ}(x^n)} \right). \tag{31}$$

Let $\mathbb{I}_{\phi^\star \neq \phi^\circ}(x^n, \theta, \Phi^M)$ be the usual indicator function of the event $[\phi^\star \neq \phi^\circ]$. Further, let

$$B(\theta, \Phi^M) \triangleq \sum_{x^n \in \mathcal{A}^n} \mu_\theta(x^n) \mathbb{I}_{\phi^\star \neq \phi^\circ}(x^n, \theta, \Phi^M). \tag{32}$$

Lemma 2.3.2. *If $M = O\left(n^{\frac{d}{2}(1-\epsilon)}\right)$ for some $\epsilon > 0$, then for all Φ^M and $\lambda > 0$, we have*

$$\mathbf{P}_\theta \left[B(\theta, \Phi^M) \geq \lambda \right] = o\left(\frac{1}{n^{\frac{d}{2}\epsilon}} \right). \tag{33}$$

Proof. See Appendix A.1. □

Lemma 2.3.2 lets us deal with ϕ° instead of $\phi^\star$ in the rest of our treatment. Next, we use a probabilistic treatment in order to bound $D_n(\mu_\theta || \mu_{\phi^\circ})$ for a certain fraction of the source parameters. We assume that the parameter vector θ follows the capacity achieving Jeffreys' prior. As discussed earlier, this distribution is particularly interesting since it results in uniform convergence of redundancy over the space of the parameter vectors and hence it achieves the average minimax redundancy [47, 90].

In order to bound the average redundancy $R_n(l_n^{2p}, \theta)$, in the following, we find an upper bound on the Lebesgue measure of the volume of the event defined by $\frac{1}{n}D_n(\mu_\theta || \mu_{\phi^\circ}) < \delta$ in the d-dimensional space of θ. Since $\phi^\circ \in \Phi^m$, the total probability measure of the volume defined by $\min_{\phi_i \in \Phi^m} \frac{1}{n}D_n(\mu_\theta || \mu_{\phi_i}) < \delta$ would be upper bounded as well. This represents the probability of the event that the source has a small redundancy. This enables us to compute the desired upper bound on the probability measure of the sources with $R_n(l_n^{2p}, \theta) \leq \delta$.

Lemma 2.3.3. *Assume that the parameter vector θ follows Jeffreys' prior. Then,*

$$\mathbf{P}_\theta \left[\frac{1}{n} D_n(\mu_\theta || \mu_{\phi_i}) \leq \delta \right] = \qquad\qquad\qquad\qquad \frac{C_d}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta} \left(\frac{2\delta}{\log e}\right)^{\frac{d}{2}} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right). \tag{34}$$

Further, we have

$$\mathbf{P}_\theta \left[\min_{\phi_i \in \Phi^m} \frac{1}{n} D_n(\mu_\theta || \mu_{\phi_i}) \leq \delta \right] \leq \qquad\qquad\qquad\qquad M \frac{C_d}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta} \left(\frac{2\delta}{\log e}\right)^{\frac{d}{2}} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right). \tag{35}$$

Proof. See Appendix A.2. □

Lemma 2.3.3 states that the probability of the event that $\frac{1}{n}D_n(\mu_\theta || \mu_{\phi_i}) < \delta$ does not depend on ϕ_i when θ follows Jeffreys' prior. Further, the probability of the event $\min_{\phi_i \in \Phi^M} \frac{1}{n}D_n(\mu_\theta || \mu_{\phi_i}) < \delta$ becomes only a function of m. In fact, it is independent of the choice of the points in Φ^m in the space of θ, as long as the points are chosen far apart so that the regions covered by each data point do not overlap. We are now equipped to prove the main result given in Theorem 2.3.1.

Proof of Theorem 2.3.1. Using Lemma 2.3.2, we can rewrite (17) as:

$$R_n(l_n^{2p}, \theta) = \min_{\phi_i \in \Phi^m} \left\{ \hat{l}_n(\phi_i) + D_n(\mu_\theta || \mu_{\phi_i}) \right\} \;\; w.h.p. \tag{36}$$

Hence,

$$\mathbf{P}_\theta\left[\frac{R_n(l_n^{2p},\theta)}{\frac{d}{2}\log n}\leq 1-\epsilon\right] \tag{37}$$

$$\leq \mathbf{P}_\theta\left[\min_{\phi_i\in\Phi^m}\left\{\hat{l}_n(\phi_i)+D_n(\mu_\theta||\mu_{\phi_i})\right\}\leq(1-\epsilon)\frac{d}{2}\log n\right] \tag{38}$$

$$\leq \sum_{i=1}^{M}\frac{C_d}{\int|\mathcal{I}_n(\theta)|^{\frac{1}{2}}d\theta}\left(\frac{2\delta(\phi_i)}{\log e}\right)^{\frac{d}{2}}\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right). \tag{39}$$

The last inequality is obtained using Lemma 2.3.3. Here, $\delta(\phi_i)$ is given by

$$\delta(\phi_i)=(1-\epsilon)\frac{d}{2n}\log n-\frac{\hat{l}_n(\phi_i)}{n}. \tag{40}$$

The inequality in (39) holds for all values of M and all length functions $\hat{l}_n$ corresponding to π. We can minimize the right hand side to find an upper bound that is independent of the value of M and the length function $\hat{l}_n$.:

$$\mathbf{P}_\theta\left[\frac{R_n(l_{2p},\theta)}{\frac{d}{2}\log n}\leq 1-\epsilon\right]$$

$$\leq \min_M\min_{\hat{l}_n}\left\{\sum_{i=1}^{M}\frac{C_d}{\int|\mathcal{I}_n(\theta)|^{\frac{1}{2}}d\theta}\left(\frac{2\delta(\phi_i)}{\log e}\right)^{\frac{d}{2}}\right\} \tag{41}$$

upto a multiplicative constant $\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right)$. Carrying out the inner minimization in (39) leads to $\hat{l}_n(\phi_i)=\log M$ and hence $\delta_{\text{op}}=(1-\epsilon)\frac{d}{2n}\log n-\frac{\log M}{n}$, which in turn leads to the following:

$$\mathbf{P}_\theta\left[\frac{R_n(l_{2p},\theta)}{\frac{d}{2}\log n}\leq 1-\epsilon\right]$$

$$\leq \min_M\left\{M\frac{C_d}{\int|\mathcal{I}_n(\theta)|^{\frac{1}{2}}d\theta}\left(\frac{2\delta_{\text{op}}}{\log e}\right)^{\frac{d}{2}}\right\}\left(1+O\left(\frac{1}{\sqrt{n}}\right)\right). \tag{42}$$

Now, by carrying out the outer minimization we find that the optimal value of M, denoted by M_{op} is given by

$$M_{\text{op}}=\frac{n^{\frac{d}{2}(1-\epsilon)}}{e^{\frac{d}{2}}}. \tag{43}$$

First, note that $M_{\text{op}}=O\left(n^{\frac{d}{2}(1-\epsilon)}\right)$ and hence Lemma 2.3.2 applies. Further, we are interested in points $\theta\in\Lambda$ such that $\mathbf{E}\log\left(\frac{\mu_\theta(x^n)}{\mu_{\phi^\star}(x^n)}\right)=O(1)$. Hence, it is straightforward to see that the error term due to using ϕ° instead of $\phi^\star$ can be extracted in the $1 + o(1)$ error term. By using M_{op} in (41), the desired result in Theorem 2.3.1 is obtained. □

2.3.2 Average Minimax Redundancy of Ordinary Two-Part Codes

In this section, we characterize the average minimax redundancy for two-part codes.

Theorem 2.3.4. *In the universal compression of the family of parametric sources $\mathcal{P}^d_\Lambda$, the average minimax redundancy of two-part codes is obtained by*

$$\bar{R}^{2p}_n = \bar{R}_n + g(d) + O\left(\frac{1}{n}\right). \tag{44}$$

Here, $\bar{R}_n$ is the average minimax redundancy defined in (9) and $g(d)$ is the penalty term attributed to the ordinary two-part scheme given by

$$g(d) = \log \Gamma\left(\frac{d}{2}+1\right) - \frac{d}{2}\log\left(\frac{d}{2e}\right). \tag{45}$$

Proof. Let $F(n,d,\theta,\epsilon) \triangleq \frac{C_d}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta}\left(\frac{d}{en^{\epsilon}}\right)^{\frac{d}{2}}$. Denote $R_\epsilon \triangleq (1-\epsilon)\frac{d}{2}\log n$ as a redundancy level. Then, according to Theorem 2.3.1, for any ϵ such that $1 - F(n,d,\theta,\epsilon) > 0$, R_ϵ is a lower bound on the maximum redundancy. This is due to the fact that $P[R_n(l_n,\theta) > R_\epsilon] > 0$, i.e., there exists at least one parameter θ such that $R_n(l_n,\theta) > R_\epsilon$. Moreover, note that the average minimax redundancy is achieved when the parameters follow Jeffreys' prior [24, 47]. Therefore, the maximum redundancy in our case is the average minimax redundancy and we have $\bar{R}^{2p}_n > R_\epsilon$. Note that as described in Section 2.3.1, the lower bound in Theorem 2.3.1 is tight and achievable. If we minimize ϵ (maximize R_ϵ) with the constraint that $F(n,d,\theta,\epsilon) < 1$, we get the tightest lower bound on the average minimax redundancy as

$$\bar{R}^{2p}_n = \frac{d}{2}\log n - \log C_d + \log \int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta - \frac{d}{2}\log\left(\frac{d}{e}\right), \tag{46}$$

Theorem 2.3.4 is inferred if C_d is substituted from (27) in (46). □

2.3.3 Normalized Two-Part Code Redundancy

Thus far, we established a lower bound on the average redundancy for the universal compression of the family of parametric sources when the coding scheme is restricted to the ordinary two-part codes. Now, we relax this constraint and obtain the lower bound on the average redundancy of universal compression for normalized two-part coding.

Theorem 2.3.5. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of parametric sources $\mathcal{P}^d_\Lambda$. Let ϵ be a real number. Then,*

$$\mathbf{P}_\theta \left[\frac{R_n(l_n^{n2p}, \theta)}{\frac{d}{2}\log n} \leq 1 - \epsilon \right] \leq \frac{1}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta} \left(\frac{2\pi e}{n^\epsilon} \right)^{\frac{d}{2}} f(n), \tag{47}$$

where $f(n) = 1 + o(1)$.

Note that it is straightforward to deduce Theorem 2.1.3 for the case of normalized two-part codes from Theorem 2.3.5 for $\epsilon > 0$ by letting $n \to \infty$. Before we prove the theorem, we state a corollary that transforms it into a form similar to Merhav and Feder's theorem (Theorem 2.1.5 in this dissertation).

Corollary 2.3.6. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of parametric sources $\mathcal{P}^d_\Lambda$. Let ϵ be a real number. Then,*

$$\mathbf{P}_\theta \left[\frac{R_n(l_n^{n2p}, \theta)}{\bar{R}_n} \leq 1 - \epsilon \right] \leq 2^{-\epsilon \bar{R}_n} f(n), \tag{48}$$

where $f(n) = 1 + o(1)$ and $f(n) < e$.

Proof. It is straightforward to transform the result of Theorem 2.3.5 into this form by using the proper ϵ. The fact that $f(n) < e$ is deduced from Theorem 2.1.5. □

Corollary 2.3.6 provides with a tight converse on the universal compression of parametric sources.

The key in the proof of Theorem 2.3.5 is the following lemma that puts an upper bound on the saving achieved by using the normalized two-part codes.

Lemma 2.3.7. *The penalty term in the average redundancy of the two-part coding is upper bounded as*

$$R_n(l_n^{2p}, \theta) - R_n(l_n^{n2p}, \theta) \leq g(d) + O\left(\frac{1}{n}\right). \tag{49}$$

Proof. See Appendix A.3 □

Lemma 2.3.7 states that the difference between the average redundancy of two-part codes with that of the normalized two-part codes is no larger than $g(d)$, which is the difference between the minimax redundancy of the two-part codes with that of the normalized two-part codes. We may now prove Theorem 2.3.5.

Proof of Theorem 2.3.5. First note that according to Lemma 2.3.7, we have

$$R_n(l_n^{2p}, \theta) \leq R_n(l_n^{n2p}, \theta) + g(d) + O\left(\frac{1}{n}\right). \tag{50}$$

Hence,

$$\begin{aligned} \mathbf{P}_\theta &\left[\frac{R_n(l_n^{n2p}, \theta) + g(d)}{\frac{d}{2}\log n} \leq 1 - \hat{\epsilon}\right] \\ &\leq \mathbf{P}_\theta \left[\frac{R_n(l_n^{2p}, \theta)}{\frac{d}{2}\log n} \leq 1 - \hat{\epsilon}\right] \\ &\leq \frac{C_d}{\int |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta} \left(\frac{d}{en^{\hat{\epsilon}}}\right)^{\frac{d}{2}}, \end{aligned} \tag{51}$$

where the second inequality is due to Theorem 2.3.1, for any $\hat{\epsilon}$. Now, the desired result is then achieved if we set ϵ such that

$$(1-\epsilon)\frac{d}{2}\log n = (1-\hat{\epsilon})\frac{d}{2}\log n - g(d).$$

□

2.3.4 Two-Part Code Redundancy for Memoryless Sources

In this section, we tailor the results for the class of memoryless sources due to their importance. Although, the material in this section is directly resulted from the more general results on the class of parametric sources, alternative proofs for the material in this section may be found in [15]. Let $\mathcal{M}_m^{|\mathcal{A}|}$ denote the family of m-th order Markov sources with alphabet size $|\mathcal{A}|$. Consequently, $\mathcal{M}_0^{|\mathcal{A}|}$ will be used to denote the family of memoryless sources. Let $k = |\mathcal{A}|$ be the size of the alphabet $\mathcal{A}$. The parameter vector $\theta = (\theta_1, ..., \theta_k)$ can be described as $\theta_j = \mathbf{P}[X = \alpha_j]$ and $\sum_j \theta_j = 1$. Note that the parameters lie in a $(k-1)$-dimensional simplex, i.e., $d = k-1$. Let r_i count the appearance of symbol α_i in sequence x^n. Let f_i denote the empirical mass function for the symbol α_i, i.e., $f_i = r_i/n$. Then, the probability measure μ_θ over a memoryless source with parameter vector θ is

$$\mu_\theta(x^n) = \mathbf{P}[X^n = x^n|\theta] = \prod_{i=1}^{k} \theta_i^{r_i}. \tag{52}$$

Further, let $\phi^\star = (\phi_1^\star, ..., \phi_k^\star) \in \Phi^m$ denote the optimal estimated point for the sequence x^n. Then, the probability measure defined by $\phi^\star$ is

$$\mu_{\phi^\star}(x^n) = \prod_{i=1}^{k} \phi_i^{\star r_i}. \tag{53}$$

In the following, we state the main results on the compression of finite-alphabet memoryless sources:

Corollary 2.3.8. *Consider the universal compression of the family of memoryless sources $\mathcal{M}_0$. Assume that the parameter vector θ follows Jeffreys' prior. Let ϵ be a real number. Then,*

$$\mathbf{P}_\theta \left[\frac{R_n(l_n^{2p}, \theta)}{\frac{k-1}{2}\log n} \leq 1 - \epsilon \right] \leq \left(\frac{k-1}{en^\epsilon} \right)^{\frac{k-1}{2}} D_k, \tag{54}$$

where

$$D_k = \frac{\Gamma\left(\frac{k}{2}\right)}{\Gamma\left(\frac{k+1}{2}\right)} \sqrt{\frac{1}{\pi}}. \tag{55}$$

Note that $D_k \approx \sqrt{\frac{2}{k\pi}}$ for $k \gg 2$. Corollary 2.3.8 is a direct consequence of Theorem 2.3.1. In the case of memoryless sources, Jeffreys' prior for the parameter vector θ is given by

$$p_J(\theta) = \frac{\Gamma\left(\frac{k}{2}\right)}{\Gamma\left(\frac{1}{2}\right)^k} \prod_{j=1}^{k} \frac{1}{\sqrt{\theta_j}}, \tag{56}$$

where $\Gamma(\cdot)$ is Euler's gamma function defined in (25). This is in fact the $(\frac{1}{2}, ..., \frac{1}{2})$ Dirichlet distribution. Further, the square root of the determinant of the Fisher information matrix may be analytically obtained as

$$\int_{\Lambda} |\mathcal{I}_n(\theta)|^{\frac{1}{2}} d\theta = \frac{\Gamma\left(\frac{1}{2}\right)^k}{\Gamma\left(\frac{k}{2}\right)}. \tag{57}$$

This expression enables us to further simplify the main results in Theorem 2.3.4 for memoryless sources as the following.

Corollary 2.3.9. *Consider the universal compression of the family of memoryless sources $\mathcal{M}_0$. Then, the average minimax redundancy of two-part codes is obtained by*

$$\bar{R}_n^{2p} = \bar{R}_n + \log \Gamma\left(\frac{k+1}{2}\right) - \frac{k-1}{2} \log\left(\frac{k-1}{2e}\right) + O\left(\frac{1}{n}\right), \tag{58}$$

where $\bar{R}_n$ is the average minimax redundancy for memoryless sources given by [24] as

$$\bar{R}_n = \frac{k-1}{2} \log\left(\frac{n}{2\pi}\right) + \log\left(\frac{\Gamma\left(\frac{1}{2}\right)^k}{\Gamma\left(\frac{k}{2}\right)}\right) + O\left(\frac{1}{n}\right). \tag{59}$$

Corollary 2.3.9 gives the extra redundancy due to the two-part coding of the memoryless sources.

In the following, we present Theorem 2.3.5 for the special case of the memoryless sources.

Corollary 2.3.10. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of memoryless sources $\mathcal{M}_0$. Let ϵ be a real number. Then,*

$$\mathbf{P}_\theta \left[\frac{R_n(l_n^{n2p}, \theta)}{\frac{k-1}{2} \log n} \leq 1 - \epsilon \right] \leq \frac{\Gamma\left(\frac{k}{2}\right)}{\sqrt{\pi}} \left(\frac{2}{n^\epsilon}\right)^{\frac{k-1}{2}}. \tag{60}$$

2.3.5 Achievability

It is not difficult to argue that the converse bound derived so far is indeed achievable. We only need to demonstrate that a set of non-overlapping d-dimensional ellipsoids (weighted by the inverse of the least favorable Jeffreys' prior) can be packed into the d-dimensional space Λ of the parameter vectors. Since the volume of these ellipsoids tends to zero as they approach the boundaries of the space, we can move the ellipsoids toward the boundaries and ensure that they do not overlap whilst covering the same measure under the least favorable prior. Therefore, by choosing such covering of the space we can indeed make sure that the bounds are achieved.

Please note that in order to achieve the average minimax redundancy, we will need to cover the space with ellipsoids [14]. Although sphere packing [9, 53] and sphere covering [25] are distinct problems, it turns out that in our problem they have similar limiting behaviors when a set of non-overlapping d-dimensional ellipsoids weighted by the inverse of the least favorable Jeffreys' prior are involved.

2.4 Elaboration on the Results

In this section, we elaborate on the significance of our results.

2.4.1 Redundancy in Finite-Length Sequences with Small d

We demonstrate that the average minimax redundancy underestimates the performance of source coding in the small to moderate length n for sources with small d. In Figures 1 and 2, the x-axis denotes a fraction P_0 and the y-axis represents a redundancy level R_0. The solid curves demonstrate the derived lower bound on the average redundancy of the normalized two-part codes R_0 as a function of the fraction P_0 of the sources with redundancy larger than R_0, i.e., $\mathbf{P}_\theta[R_n(l_n^{n2p}, \theta) \geq R_0] \geq P_0$. In other words, the pair (R_0, P_0) on the redundancy curve means that at least a fraction P_0 of the sources that are chosen from Jeffreys' prior have an average redundancy that is

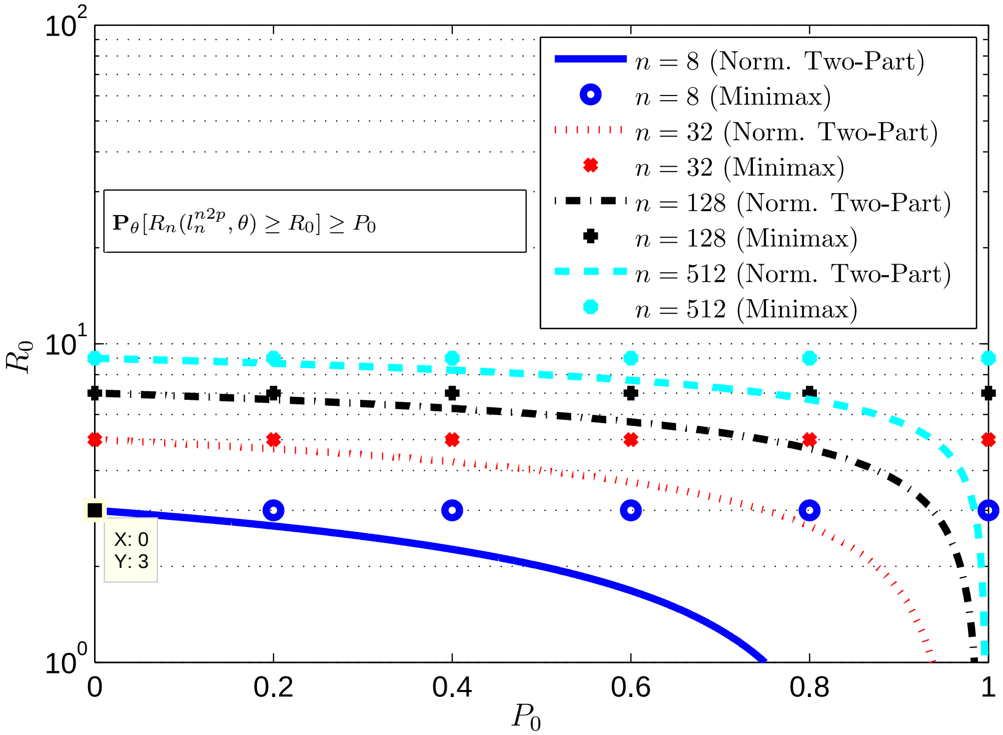

Figure 1: Average redundancy of the normalized two-part codes (*Norm. Two-Part*) and the average minimax redundancy (*Minimax*) as a function of the fraction of sources P_0 whose redundancy satisfies $R_n(l_n^{n2p}, \theta) > R_0$. Memoryless source $\mathcal{M}_0^3$ with $k = 3$ and $d = 2$.

greater than R_0. Note that the unknown parameter vector is chosen using Jeffreys' prior in all of the examples.

First, we consider a ternary memoryless information source denoted by $\mathcal{M}_0^3$. Let k be the alphabet size, where $k = 3$. This source may be parameterized using two parameters, i.e., $d = 2$. In Figure 1, our results are compared to the average minimax redundancy, i.e., $\bar{R}_n$ from (9). Since the normalized two-part codes achieve the minimax redundancy, $\bar{R}_n$ is in fact the average minimax redundancy for the normalized two-part codes ($\bar{R}_n^{n2p}$) as well. The results are presented in bits. As shown in Figure 1, at least 40% of ternary memoryless sequences of length $n = 32$ ($n = 128$) may not be compressed beyond a redundancy of 4.26 (6.26) bits. Also, at least 60% of ternary memoryless sequences of length $n = 32$ ($n = 128$) may not be compressed beyond

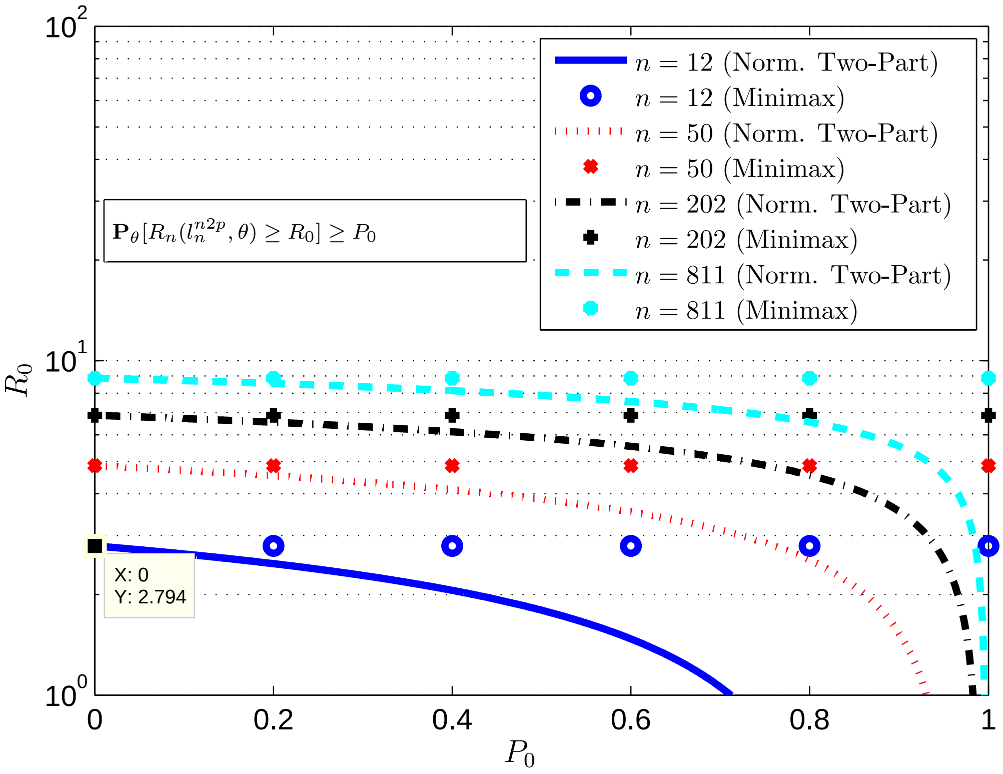

Figure 2: Average redundancy of the normalized two-part codes (*Norm. Two-Part*) and the average minimax redundancy (*Minimax*) as a function of the fraction of sources P_0 whose redundancy satisfies $R_n(l_n^{n2p}, \theta) > R_0$. First-order Markov source $\mathcal{M}_1^2$ with $k = 2$ and $d = 2$.

a redundancy of 3.67 (5.68) bits. Note that as $n \to \infty$, the average redundancy approaches the average minimax redundancy for most sources.

Next, let $\mathcal{M}_1^2$ denote a binary first-order Markov source ($d = 2$). We present the finite-length compression results in Figure 2 for different values of sequence length n. The values of n are chosen such that they are almost $\log(3)$ times the values of n for the ternary memoryless source in the first example. This choice has been made to equate the amount of information in the two sequences from $\mathcal{M}_0^3$ and $\mathcal{M}_1^2$ allowing a fair comparison. For example, a sequence of length $n = 8$ from source $\mathcal{M}_0^3$, consisted of 8 ternary symbols, is equivalent to $8\log(3)$ bits of information that is almost equivalent to 12 bits in $\mathcal{M}_1^2$.

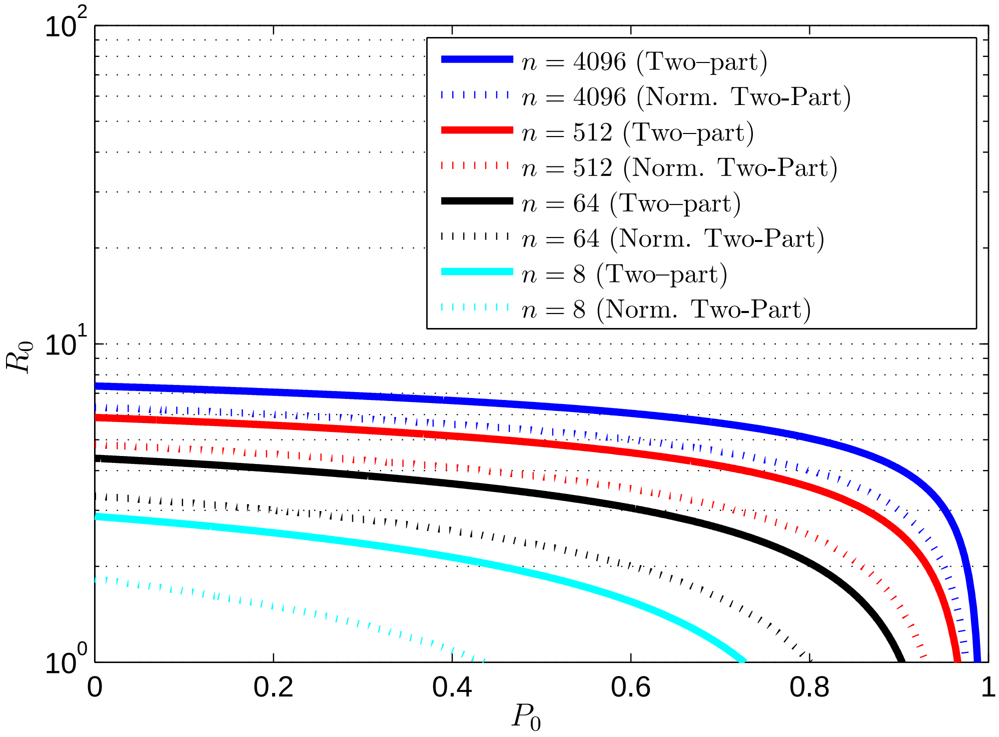

Figure 3: Average redundancy of the two-part codes (solid) vs average redundancy of the normalized two—stage codes (dotted) as a function of the fraction of sources P_0. Memoryless source $\mathcal{M}_0^2$ with $k = 2$ and $d = 1$.

Figure 2 shows that the average minimax redundancy of the normalized two-part codes for the case of $n = 12$ is given as $\bar{R}_{12} \approx 2.794$ bits. Comparing Figure 1 with Figure 2, we conclude that the average redundancy of universal compression for a binary first-order Markov source is very similar to that of the ternary memoryless source, suggesting that d is the most important parameter in determining the average redundancy of finite-length sources. This subtle difference becomes even more negligible as $n \to \infty$ since the dominating factor of redundancy for both cases approaches to $\frac{d}{2} \log n$.

Further, as demonstrated in Figures 1 and 2, there is a significant gap between the known result by the average minimax redundancy and the finite-length results obtained in this chapter when a high fraction P_0 of the sources is concerned. The bounds derived in this chapter are tight, and hence, for many sources the average

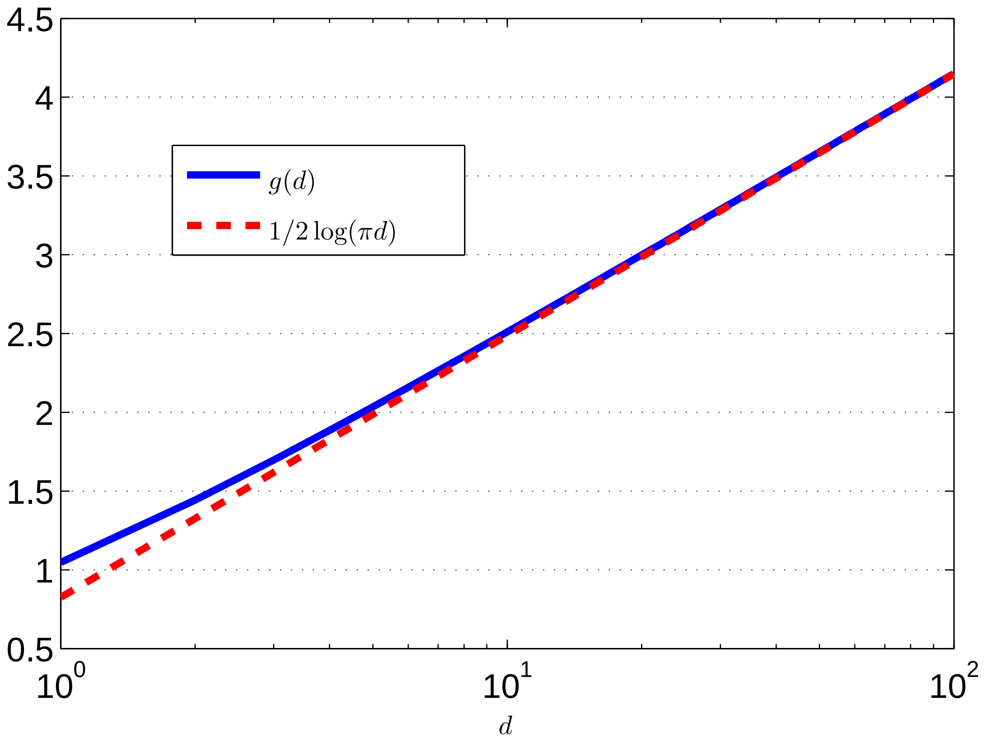

Figure 4: The extra redundancy incurred due to the two-part assumption on the code as a function of d.

minimax redundancy overestimates the average redundancy in universal source coding of finite-length sequences where the number of the parameters d is small. In other words, the compression performance of a high fraction of finite-length sources would be better than the estimate given by the average minimax redundancy.

2.4.2 Two-Part Codes Vs Normalized Two-Part Codes

Next, we compare the finite-length performance of the two-part codes with that of the normalized two-part codes on the class of binary memoryless source $\mathcal{M}_0^2$ with $k = 2$ ($d = 1$). The results are presented in Figure 3. The solid line and the dotted curves demonstrate the lower bound for the two-part codes and the normalized two-part codes, respectively. As can be seen, the gap between the achievable compression using two-part codes and that of the normalized two-part codes constitutes a significant fraction of the average redundancy for small n. For a Bernoulli source, the average

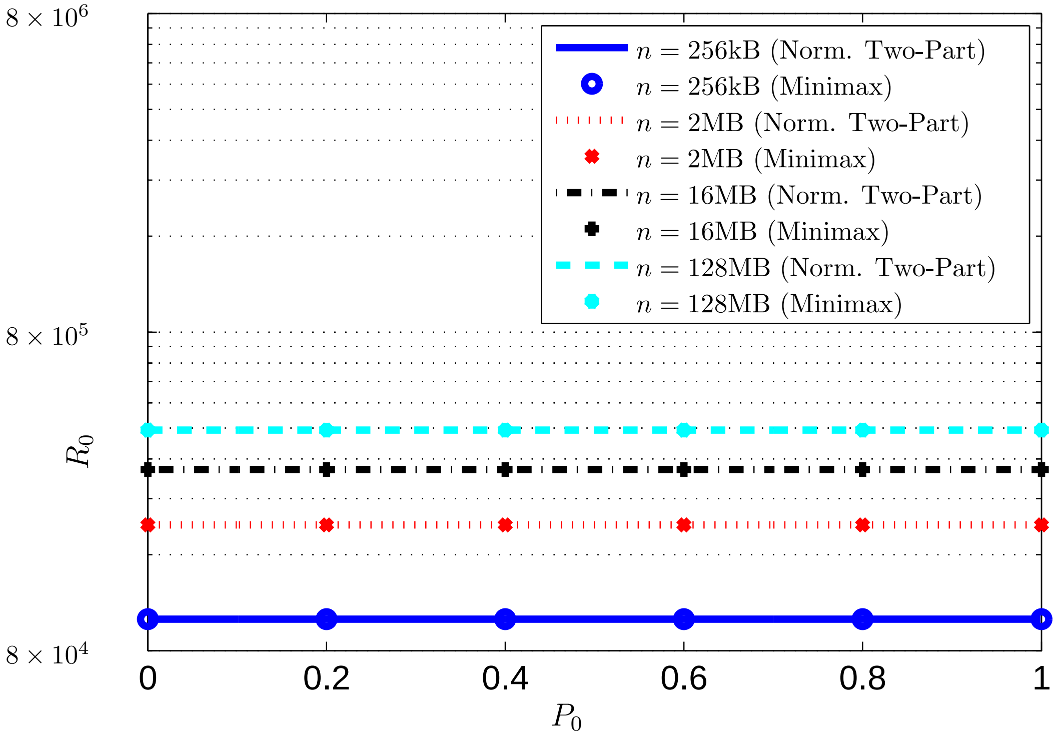

Figure 5: Average redundancy of the normalized two-part codes (*Norm. Two-Part*) and the average minimax redundancy (*Minimax*) as a function of the fraction of sources P_0 whose redundancy satisfies $R_n(l_n^{n2p}, \theta) > R_0$. First-order Markov source with $k = 256$ and $d = 65280$. The sequence length n is measured in bytes (B).

minimax redundancy of the two-part code is given in (58) as

$$\bar{R}_n^{2p} = \bar{R}_n + \frac{1}{2}\log\left(\frac{\pi e}{2}\right) \approx \bar{R}_n + 1.048. \quad (61)$$

The average minimax redundancy of two-part codes for the case of $n = 8$ is given as $\bar{R}_8^{2p} \approx 2.86$ bits while that of the normalized two-part codes is $\bar{R}_8 \approx 1.82$. Thus, the two-part codes incur an extra compression overhead of more than 50% for $n = 8$.

In Theorem 2.3.4, we derived that the extra redundancy $g(d)$ incurred by the two-part assumption. We further use Stirling's approximation for sources with large number of parameters in order to show the asymptotic behavior of $g(d)$ as $d \to \infty$. That is, asymptotically, we have

$$g(d) = \frac{1}{2}\log(\pi d) + o(1). \quad (62)$$

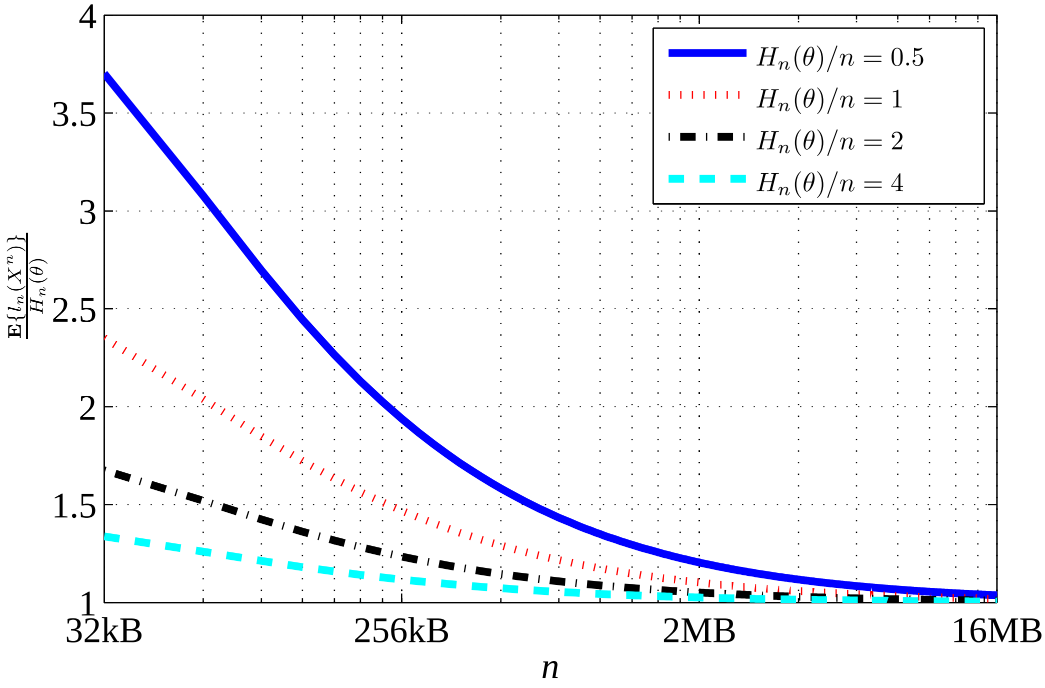

Figure 6: The Lower bound on compression for at least 95% of the sources as a function of sequence length n vs. the entropy rate $H_n(\theta)/n$.

Note that $o(1)$ denotes a function of d and not n here. As demonstrated in Figure 4, $g(d)$ is increasing logarithmically with d as $d \to \infty$. Finally, we must note that the main term of redundancy in $\bar{R}_n$ is $\frac{d}{2}\log n$, which is linear in d, but the penalty term $g(d)$ is logarithmic in d. Hence, the adverse impact of the two-part assumption becomes negligible for the families of sources with larger d.

2.4.3 Average Redundancy for Sources with Large d

In this section, we validate our results that the average minimax redundancy provides a good estimate on the achievable compression for most sources when d is large for sufficiently large n. We consider a first-order Markov source with alphabet size $k = 256$. We intentionally picked this alphabet size as it is a common practice to use the byte as a source symbol. This source may be represented using $d = 256 \times 255 = 62580$ parameters. In Figure 5, the achievable redundancy is demonstrated for four different values of n. Here, again the redundancy is measured in bits but the sequence length in this example is presented in bytes. The curves are almost flat when d and n are very large validating our results that. We further observe that for $n = 256$kB, we have $R_n(l_n, \theta) \geq 100,000$ bits for most sources. Further, the extra redundancy due to the two-part coding is equal to $g(d) \approx 8.8$ bits, which is a negligible fraction of the average redundancy of $100,000$ bits. This demonstrates that as the number of source parameters grow, the minimax redundancy well estimates the performance of the source coding.

2.4.4 Significance of Redundancy in Finite-Length Compression

One of the results of this chapter is to characterize the significance of redundancy in finite-length compression. Figure 6 demonstrates the average number of bits per symbol normalized to the entropy of the sequence for different values of entropy rates required to compress the class of the first-order Markov sources . In this figure, the dashed red curve demonstrates the lower bound on the achievable compression rate for at least 95% of sources with entropy rate of 1 bit per source symbol (per byte), i.e., at least 95% of the sources from this class may not be compressed with a redundancy smaller than the value given by the curve. We consider these sources since many practical sources have an entropy rate that is smaller than 1 bit per source symbol. As can be seen, the compression overhead is 38%, 16%, 5.5%, 1.7%, and 0.5% for sequences of lengths 256kB, 1MB, 4MB, 16MB, and 64MB, respectively. Hence, we conclude that redundancy may be significant for the compression of low entropy sequences of length up to 1MB. On the other hand, redundancy is negligible for sequences of length 64MB and higher. This shows that the redundancy is significant in the compression of small to medium length sequences with large number of parameters.

2.5 Conclusion

In this chapter, the average redundancy rate of universal coding schemes on parametric sources in the *finite*-length regime was investigated. A lower bound on the probability of the event that an information source chosen using Jeffreys' prior from the family of parametric information sources is not compressible beyond any certain fraction of the average minimax redundancy was derived. This result may be viewed as the finite-length extension of the existing asymptotic results. It was demonstrated that the average minimax redundancy underestimates the performance of source coding in the small to moderate length sequences for sources with small number of parameters. The performance of two-part codes was compared with normalized two-part codes. It was shown that the penalty term of the ordinary two-part coding is negligible for sources with large d as well as for the sequences of sufficient lengths. Further, as the number of source parameters grows very large, the average minimax redundancy provides an accurate estimate for the performance of the source coding. Finally, our results collectively demonstrate that the redundancy is significant in the universal compression of small length sequences with large number of source parameters, such as the network packets.

CHAPTER III

NETWORK COMPRESSION VIA MEMORY

The presence of considerable amounts of correlation in network traffic data has motivated the employment of correlation elimination techniques for network traffic data [92, 4, 6, 5, 80]. The present correlation elimination techniques are mostly based on (content) caching mechanisms used by solutions such as web-caching [39], CDNs [58], and P2P applications [57]. However, several experiments confirm that the caching approaches, which take place at the application layer, do not efficiently leverage the network redundancy which exists mostly at the packet level [92, 4, 6, 5]. To address these issues, a few recent studies have considered ad-hoc methods such as packet-level Redundancy Elimination (RE) in which redundant transmissions of segments of a packet that are seen in previously sent packets are avoided [6, 5]. However, these techniques are limited in scope and can only eliminate exact duplications from the segments of the packets.

Universal compression schemes may also be considered as potential *end-to-end* correlation elimination techniques for network traffic data.[1] However, as characterized in Chapter 2, universality imposes an inevitable redundancy, which is due to the learning of source statistics. In Section 3.1, we will demonstrate that the universal compression of finite-length network packets (up to hundreds of kilobytes) incurs a significant redundancy (i.e., overhead). This places a strict performance limit for many practical applications including compression of packets from a single parametric source in the network. In other words, such techniques require infinite length data to effectively

[1] Please note that by end-to-end universal compression we mean that the encoding of the packet is performed at the server and the decoding of the packet is performed at the client without using the intermediate nodes in the network, and hence, the name end-to-end.

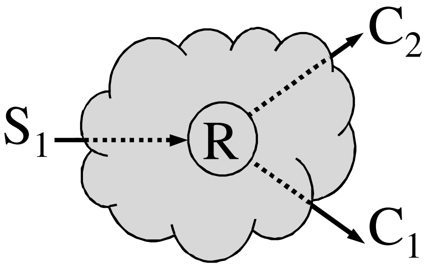

Figure 7: The basic scenario of single-source memory-assisted compression.

remove redundancy. Further, the end-to-end traditional information theoretic techniques would only attempt to deal with the first type of redundancy mentioned above (i.e., redundancy within a content) and lacks the structure to leverage the second dimension of redundancy.

In this chapter, *network compression via memory* is proposed. The basic premise of the network compression relies on the rational that network elements, such as routers or other intermediate nodes, can be enabled to memorize the traffic and learn about network source contents as they forward the packets. This knowledge about source contents is then used by a properly designed memory-assisted compression mechanism to represent new contents more efficiently (i.e., smaller codewords). In a nutshell, equipping the network with network memory elements will enable memorization of the source traffic as it flows naturally (or by design) through the network. As such, memory enabled nodes can learn the source data statistics which can then be used (as side information) toward reducing the cost of describing the source data statistics in compression. The idea is based on the fact that training data can be useful in decreasing the redundancy in universal compression (cf. [47, 45, 93] and the references therein). We theoretically formulate the packet memorization gain problem and investigate the fundamental limits of network compression via network memory. The objective of this chapter is to study the network compression via memory involving a single source to leverage the two types of redundancy mentioned earlier. This is in contrast to the Slepian-Wolf coding that targets the spatial redundancy between distributed information sources [79, 71, 56].[2]

The rest of this chapter is organized as follows. In Section 3.1, the concept of network compression is introduced and the necessary background is presented. In Section 3.2, the network compression problem setup is defined. In Section 3.3, the memorization gain is defined and our main results on the memorization gain are provided. In Section 3.4, the benefits of the memory-assisted compression are demonstrated in a network through a case study. In Section 3.6, the technical analysis of the main results of the chapter is presented. Finally Section 3.7 concludes this chapter.

3.1 Basic Concept of Network Compression

The network compression module resides, by design, at the network layer. We describe network compression in the most basic network scenario depicted in Figure 7. As described in Section 2.1, we use the notation $x^n = (x_1, ..., x_n)$ to present a sequence of length n, where each symbol x_i is from the alphabet $\mathcal{A}$. For example, for an 8-bit alphabet that has 256 symbols, each x_i is a byte. Note also that x^n denotes a single packet at the network layer. We assume that, as shown in Figure 7 as a basic setup, the network consists of the server S, the intermediate memory-enabled (relay or router) node M, and the clients C_1 and C_2.[3] In this scenario, S wishes to send the packet x^n to C_2. As a new client, C_2 does not have any prior (recent) communication with the server, and hence, it does not have any memory regarding the source context. However, as an intermediate (relay or router) node enabled with memory, the node M has already observed a previous sequence y^m from S while

[2] Please note that in the Slepian-Wolf coding the sequences from the distributed sources are assumed to have symbol-by-symbol correlation, which is also different from our correlation model in this work that is due to the parameter vector being unknown in the universal compression.

[3] The network also possibly contains some other relay nodes in the S-M, M-C_1 and M-C_2 paths that are not shown in the figure.

forwarding it to C_1 . Therefore, the server S would encode x^n using the memory-assisted universal compression (which uses the memory y^m) and send the resulting codeword to M. Since M has already established a common memory (y^m) with server S, it can decode the incoming codeword. The node M would then possibly re-encode the sequence x^n using traditional universal compression (without memory) before sending the new codeword to the client C_2, which can decode it using the corresponding traditional universal compression method. Note that since on the link from S to M, the packet x^n is compressed using memory, we expect to leverage the knowledge that is provided by the memory about the packet (x^n). We will quantify this memorization gain later in Section 3.3. We emphasize, however, that the above scenario is simplified for the clarity of the discussion. For example, the memory y^m, in practice, is the result of the concatenation of several previously forwarded packets by M from the server S in serving various other clients.

Alternatively, in the absence of the network compression, the delivery of the packet (x^n) from S to C_2 in Figure 7 is performed using the traditional compression schemes as follows. Since there is no utilization of memory at M, the source can only apply traditional universal compression to packet x^n and transmit the resulting codeword to M who simply forwards the incoming codeword to C_2. We refer to this latter method as end-to-end universal compression, which does not utilize memory. Obviously, there is potential performance benefits offered by network compression (due to memory) on the link from S to M that is missed in the second method (end-to-end compression). Note that on the link from M to C_2, both network compression and end-to-end compression have identical performance. Therefore, it is clear from the above discussion that to characterize the benefits offered by network compression (which uses memory-assisted compression), we need to quantify the performance advantage of memory-assisted universal compression over traditional universal compression of a packet on the link from the server to the memory element (i.e., from

S to M in Figure 7). In the nutshell, as it will become clear, this advantage is due to the finiteness of the packet length and the penalty incurred by the universal compression of the finite-length sequences. Namely, we would like to answer the following questions in the above setup:

1. Would the deployment of memory in the source (encoder) and the intermediate node (decoder) provide any fundamental benefit in the universal compression of a packet from a single source?

2. If so, how does this gain vary as the packet length n and the memorized context length m change?

In the context of end-to-end compression, the performance of traditional universal compression techniques was reviewed in Chapter 2, where using a probabilistic treatment, a lower bound was provided on the probability measure of the event that a sequence of finite length n from a parametric source is compressed using the family of conditional two-part codes with a redundancy larger than a certain fraction of the average minimax redundancy. To demonstrate the significance of this result, we considered an example using a first-order Markov source with alphabet size $k = 256$. This source may be represented using $d = 256 \times 255 = 62580$ parameters. We further assume that the source entropy rate is 0.5 bit per byte ($H_n(\theta)/n = 0.5$). Please note that we will confirm this assumption shortly using real network data. Then, our result from Section 2.4 suggests that the compression overhead is more than 75% for sequences of length 256kB. Hence, we concluded that redundancy is significant in the compression of finite-length low-entropy sequences, such as the Internet traffic packets that are much shorter than 256kB. It is this redundancy that we hope to remove using the memorization technique. The compression overhead becomes negligible for very long sequences (e.g., it is less than 2% for sequences of length 64MB and above), and hence, the benefits of the memorization technique vanish as the sequence length

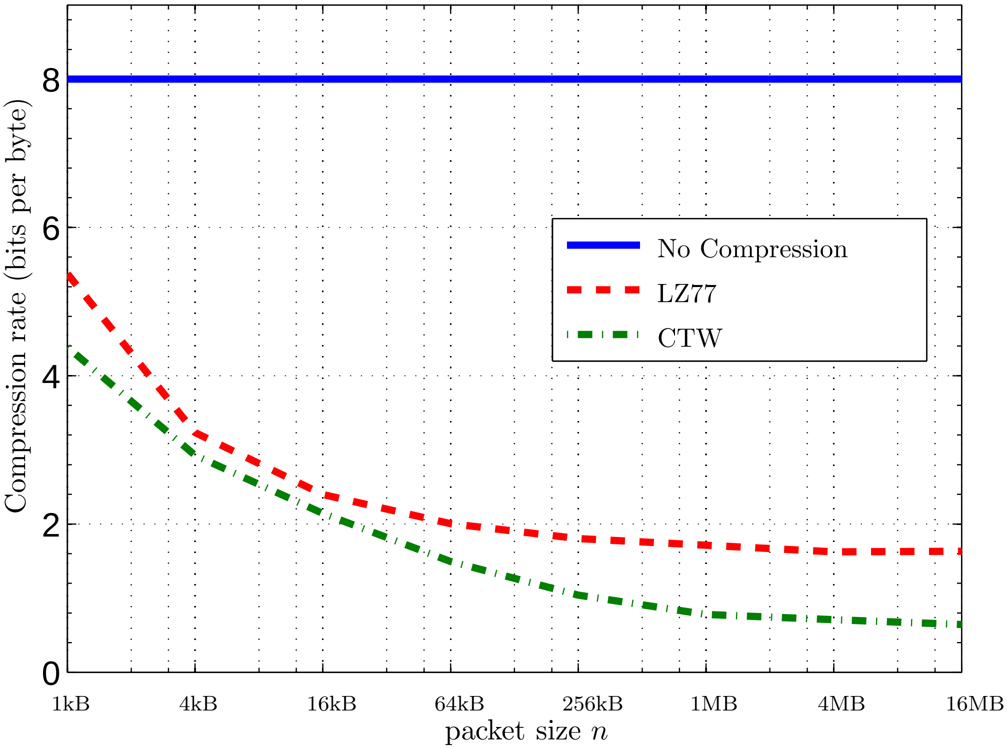

Figure 8: The compression rate for LZ77 and CTW compression algorithms on CNN server files as a function of the sequence length.

grows to infinity. This is due to the fact that the compression overhead is $O(\log n/n)$ which vanishes for large n.

Similar conclusions can be drawn for real network data tarces. To illustrate the significance of the redundancy when network packets are universally compressed, we performed several experiments using some well-known universal compression algorithms. In particular, we used packets that were gathered from CNN web server in seven consecutive days. Although we arbitrarily chose this web server for our illustration, similar conclusions can be drawn using data from other web servers. We used both Context Tree Weighting (CTW) [88] and LZ77 algorithm [94] for the compression. As shown in Figure 8, a modest performance can be achieved by universal compression when the length n of the packet to be compressed is relatively small. For example, for a packet of length n = 1kB, the compression rate is about 5 bits per byte. Note that the uncompressed packet requires 8 bits per source byte representation. We also note that as the sequence length n increases, the compression performance improves.[4] For very long sequences, the compression rate is about 0.5 bits per byte confirming our earlier assumption of $H_n(\theta)/n = 0.5$. In other words, comparing the compression performance between $n = 1\text{kB}$ and $n = 16\text{MB}$, there is a penalty of factor 10 on the compression performance (i.e., 5 as opposed to 0.5). This implies that for network packets which often have short or moderate lengths, there is a significant loss in the performance of universal compression. It is this redundancy that we wish to remove using the network memory.

3.2 Setup of the Network Compression

In this section, we attempt to formulate the network compression via memory in an information-theoretic context. To that end, we limit the scope of the network compression to the universal compression of packets from a single source. We further limit our study to the scenario, where the packets are generated by a stationary source. Evidently, the network compression module, which resides at the network layer, observes packets that cannot be solely from a single stationary source in a real-world content server in a network. Although the stationary assumption may appear a simplistic model, it still provides with useful insight to the fundamental gain offered by the network compression via memory. Further, as we have presented in [18, 70], a real content server can be potentially viewed as a mixture of several stationary sources in which we can solve the network compression via clustering of packets to different classes, where each class is associated with a distinct stationary source. Therefore, in the rest of this work, we will pretend that the network server is a single stationary source, where the source encoder (at the server) wishes to compress an individual packet. We assume that the source is a parametric information source

[4] Please note that each sequence can be thought of as the concatenation of several packets.

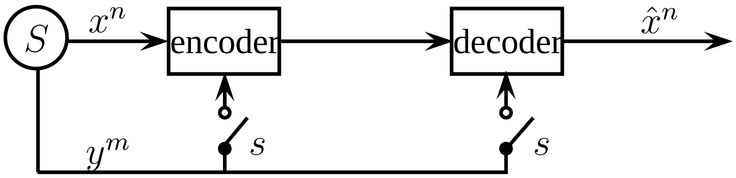

Figure 9: Abstraction of the single-source compression.

with parameter vector θ, where θ follows the least favorable Jeffreys' prior.

As explained in Section 3.1, both the encoder at S and the decoder at the memory enabled router M, in Figure 7, have another sequence y^m from the same unknown information source (i.e., the server) in common. This common sequence can be viewed as the concatenation of all previous packets that node M has forwarded from the source S to different destinations. As discussed earlier, network compression via memory and the traditional end-to-end compression only differ in the S-M link where the former utilizes memory-assisted compression for the improved performance. The model for the coding system is depicted in Figure 9. The memory-assisted compression and the traditional universal compression scenarios correspond to the switch s being *closed* and *open*, respectively. In memory-assisted compression, we wish to compress the sequence x^n when both the encoder and the decoder have access to a realization y^m of the random sequence Y^m. This setup, although very generic, can incur in many applications.

In this setup, let $c^{p_e}_{n,m} : \mathcal{A}^n \times \mathcal{A}^m \to \{0,1\}^*$ denote the memory-assisted encoding, which generates a code with the help of a side information sequence of length m. Further, denote $d^{p_e}_{n,m} : \{0,1\}^* \times \mathcal{A}^m \to \mathcal{A}^n$ as the memory-assisted decoding, which will reconstruct x^n with probability at least $(1 - p_e)$ at the decoder with the help of the same side information sequence of length m. Thus, when the sequence y^m is available to both the encoder and the decoder, the codeword associated with the compression of the sequence x^n with permissible error probability p_e is $c^{p_e}_{n,m}(x^n, y^m)$. This codeword is decoded at node M with the help of y^m with permissible error probability p_e. We have $\hat{x}^n = d^{p_e}_{n,m}(c^{p_e}_{n,m}(x^n, y^m), y^m)$. The schematic for the encoding and decoding is also illustrated in Figure 9

In order to quantify the benefits of the network compression via memory, we need to investigate the fundamental gain of the memorization in the memory-assisted universal compression of the sequence x^n over traditional universal source coding. Thus, we introduce the following two schemes.

- Ucomp (Traditional universal compression), in which a sole universal compression using $c^{p_e}_n(\cdot)$ is applied on the sequence x^n without regard to the sequence y^m. This corresponds to switch s being *open* in Figure 9.

- UcompED (Memory-assisted universal compression), in which the encoder at S and the decoder at M both have access to the memorized sequence y^m from the source S, and they use y^m to learn the statistics of the source S to better compress x^n. Thus, the compression is performed using $c^{p_e}_{n,m}(\cdot, \cdot)$. This corresponds to switch s being *closed* in Figure 9.

Let $l^{p_e}_n : \mathcal{A}^n \to \mathbb{R}$ denote the universal length function for an almost lossless code with permissible error probability p_e. In this work, we ignore the integer constraint on the length function, which results in a negligible $O(1)$ redundancy analyzed in [30, 81]. Therefore, in Ucomp coding strategy, the length of the code for the compression of the sequence x^n with error p_e is denoted by $l^{p_e}_n(x^n)$. Further, when the code is strictly lossless the length is given by $l_n(x^n) \triangleq l^0_n(x^n)$. Let $L^{p_e}_n$ denote the space of universal lossless length functions on a sequence of length n, with permissible decoding error p_e. Denote $R_n(l^{p_e}_n, \theta)$ as the expected redundancy of the almost lossless code $l^{p_e}_n$ on a sequence of length n for the parameter vector θ, defined as

$$R_n(l^{p_e}_n, \theta) = \mathbf{E} l^{p_e}_n(X^n) - H_n(\theta). \tag{63}$$

Let $\bar{R}_{\text{Ucomp}}(n)$ denote the average minimax redundancy of the Ucomp coding strategy as given by

$$\bar{R}^{p_e}_{\text{Ucomp}}(n) \triangleq \inf_{l_n^{p_e} \in L_n^{p_e}} \sup_{\theta \in \Lambda^d} R_n(l_n^{p_e}, \theta). \tag{64}$$

We denote $\bar{R}^0_{\text{Ucomp}}(n)$ as the expected minimax redundancy when the compression scheme is restricted to be strictly lossless, i.e., the case where $p_e = 0$.

In UcompED, denote $l^{p_e}_{n,m} : \mathcal{A}^n \times \mathcal{A}^m \to \mathbb{R}$ as the almost lossless universal length function for encoding a sequence of length n with permissible error probability p_e. Thus, since both the encoder and the decoder have access to a memorized sequence y^m in UcompED coding strategy, the length function for encoding the sequence x^n with error p_e is denoted by $l^{p_e}_{n,m}(x^n, y^m)$. Further, denote $L^{p_e}_{n,m}$ as the space of such lossless universal length functions. Further, let $\bar{R}_{\text{UcompED}}(n, m)$ denote the corresponding average minimax redundancy, i.e.,

$$\bar{R}^{p_e}_{\text{UcompED}}(n, m) \triangleq \inf_{l^{p_e}_{n,m} \in L^{p_e}_{n,m}} \sup_{\theta \in \Lambda^d} R_n(l^{p_e}_{n,m}, \theta). \tag{65}$$

The following trivial inequality states that the redundancy decreases when side information is available, i.e., UcompED coding strategy.

Lemma 3.2.1. *For a given permissible error probability p_e, the average minimax redundancy of UcompED coding strategy is no larger than that of Ucomp coding strategy, i.e., $\bar{R}_{UcompED}(n, m) \leq \bar{R}_{Ucomp}(n)$.*

Lemma 3.2.1 simply states that the context memorization improves the performance of the universal compression.

3.3 Fundamental Gain of Memorization

In this section, we define and characterize the fundamental gain of memorization in the different coding strategies described in Section 3.2. Roughly speaking, the gain is the ratio of the expected codeword length of the traditional end-to-end universal compression (i.e., Ucomp) to that of the universal compression with memorization (i.e., UcompED). Let $Q(n,m,\theta,p_e)$ be defined as the ratio of the expected codeword length with length function $l_n^{p_e}$ to that of $l_{n,m}^{p_e}$, i.e.,

$$Q(n,m,\theta,p_e) \triangleq \frac{\mathbf{E}l_n^{p_e}(X^n)}{\mathbf{E}l_{n,m}^{p_e}(X^n)} = \frac{H_n(\theta)+R_n(l_n^{p_e},\theta)}{H_n(\theta)+R_n(l_{n,m}^{p_e},\theta)}. \tag{66}$$

Further, let ϵ be such that $0 < \epsilon < 1$. We define $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e)$ as the gain of UcompED strategy as compared to Ucomp. That is

$$g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e) \triangleq \sup_{z\in\mathbb{R}} \left\{ z : \mathbf{P}[Q(n,m,\theta,p_e) \geq z] \geq 1-\epsilon \right\}. \tag{67}$$

In other words, $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e)$ is the fundamental gain of the memorization on a sequence of length n using UcompED coding strategy with a memory of length m for a fraction $(1-\epsilon)$ of the sources from the family $\mathcal{P}^d$, where the permissible error probability is p_e. In other words, memorization of the sequence y^m provides at least a gain $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e)$ for a fraction $(1-\epsilon)$ of the sources in the family. Our goal is to derive a lower bound on the fundamental gain of memorization $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e)$.

The following is a trivial lower bound on the memorization gain.

Lemma 3.3.1. *The fundamental gain of memorization is lower bounded by unity, i.e.,* $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e) \geq 1$.

Proof. Note that $l_n^{p_e} \in L_{n,m}^{p_e}$ and the proof trivially follows from the definition of the memorization gain. □

According to Lemma 3.3.1, the memorization does not worsen the performance of the universal compression. We stress again that the saving of memory-assisted compression in terms of flow reduction is only applicable to the links on the S-M path in Figure 7. For example, for the given context memorization gain $g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e) = g_0$, the expected number of bits needed to transfer x^n to node M is reduced from $\mathbf{E}l_n^{p_e}(X^n)$ in Ucomp to $\frac{1}{g_0}\mathbf{E}l_n^{p_e}(X^n)$ in UcompM.

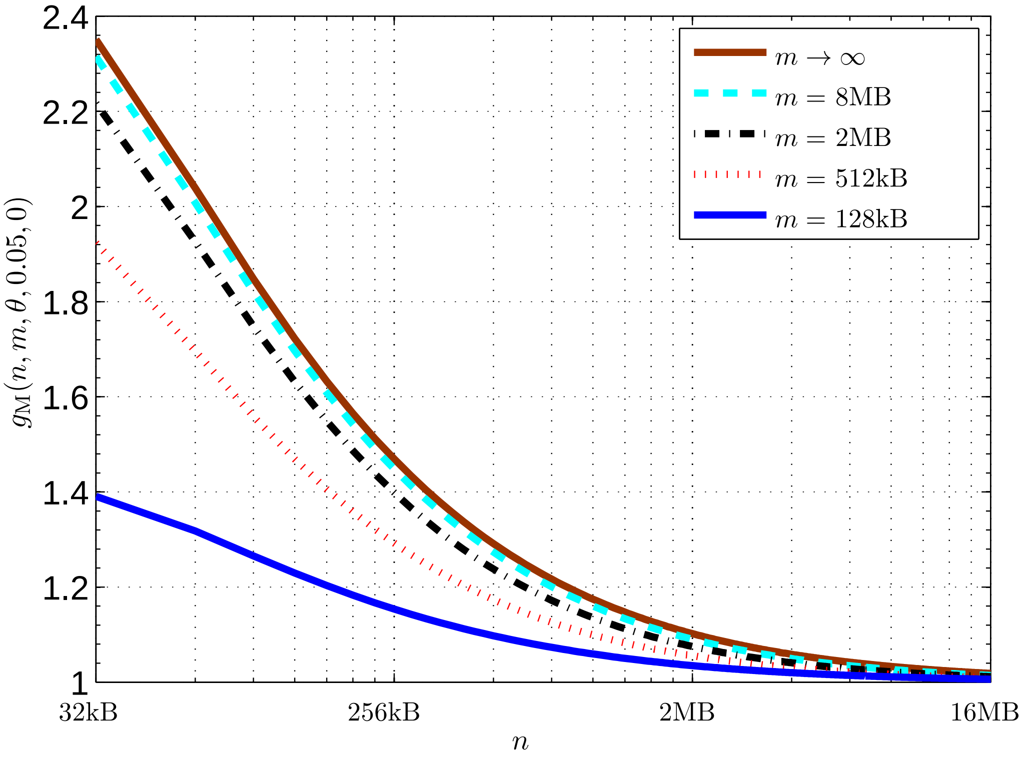

Figure 10: Theoretical lower bound on the memorization gain $g_{\mathrm{M}}(n, m, \theta, 0.05, 0)$ for the first-order Markov source with alphabet size $k = 256$ and entropy rate $H_n(\theta)/n = 0.5$.

3.3.1 Main Results on the Memorization Gain

Now, we present our main results on the fundamental gain of memorization. The proofs are deferred to Section 3.6. The next theorem characterizes the fundamental gain of memory-assisted source coding:

Theorem 3.3.2. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of parametric sources $\mathcal{P}^d$. Then,*

$$g_{\mathrm{M}}(n, m, \theta, \epsilon, p_e) \geq 1 + \frac{(1-p_e)\bar{R}^0_{Ucomp}(n) - \hat{R}_{\mathrm{M}}(n, m)}{H_n(\theta) + \hat{R}_{\mathrm{M}}(n, m)}$$

$$+ \frac{\log(\epsilon) - h(p_e) - p_e H_n(\theta) - \log e}{H_n(\theta) + \hat{R}_{\mathrm{M}}(n, m)} + O\left(\frac{1}{n\sqrt{m}}\right),$$

where $\hat{R}_{\mathrm{M}}(n,m)$ is defined as

$$\hat{R}_{\mathrm{M}}(n,m) \triangleq \frac{d}{2}\log\left(1+\frac{n}{m}\right). \tag{68}$$

and $h(p_e)$ is the binary entropy function as given by

$$h(p_e) = p_e \log\left(\frac{1}{p_e}\right) + (1-p_e)\log\left(\frac{1}{1-p_e}\right). \tag{69}$$

Let $g_{\mathrm{M}}(n,m,\theta,\epsilon,0)$ denote the fundamental gain of memorization for strictly lossless coding schemes. The following corollary bounds the memorization gain in this case.

Corollary 3.3.3. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of parametric sources $\mathcal{P}^d$. Then,*

$$\begin{aligned} g_{\mathrm{M}}(n,m,\theta,\epsilon,0) \geq 1 + \frac{\bar{R}^0_{Ucomp}(n) - \hat{R}_{\mathrm{M}}(n,m)}{H_n(\theta)+\hat{R}_{\mathrm{M}}(n,m)} \\ + \frac{\log(\epsilon) - \log e}{H_n(\theta)+\hat{R}_{\mathrm{M}}(n,m)} + O\left(\frac{1}{n\sqrt{m}}\right). \end{aligned}$$

Further, let $g_{\mathrm{M}}(n,\infty,\theta,\epsilon,p_e)$ be defined as the fundamental gain of context memorization where there is no constraint on the length of the memorized content, i.e, $g_{\mathrm{M}}(n,\infty,\theta,\epsilon,p_e) \triangleq \lim_{m\to\infty} g_{\mathrm{M}}(n,m,\theta,\epsilon,p_e)$. The following Corollary quantifies the context memorization gain for unbounded memory size.

Corollary 3.3.4. *Assume that the parameter vector θ follows Jeffreys' prior in the universal compression of the family of parametric sources $\mathcal{P}^d$. Then,*

$$\begin{aligned} g_{\mathrm{M}}(n,\infty,\theta,\epsilon,p_e) \geq 1 + \frac{(1-p_e)\bar{R}^0_{Ucomp}(n)}{H_n(\theta)} \\ + \frac{\log(\epsilon) - h(p_e) - p_e H_n(\theta) - \log e}{H_n(\theta)}. \end{aligned}$$

Next, we consider the case where the sequence length n grows to infinity. Intuitively, we would expect that the context memorization gain become negligible for the compression of long sequences. Let $g_{\mathrm{M}}(\infty, m, \theta, \epsilon, p_e) \triangleq \lim_{n\to\infty} g_{\mathrm{M}}(n, m, \theta, \epsilon, p_e)$. In the following, we claim that the context memorization does not provide any benefit when $n \to \infty$:

Proposition 3.3.5. $g_{\mathrm{M}}(n, m, \theta, \epsilon, p_e)$ *approaches unity as the length of the sequence* x^n *grows, i.e.,* $g_{\mathrm{M}}(\infty, m, \theta, \epsilon, p_e) = 1$.

Note that these results are valid for finite-length n (as long as n is large enough to satisfy the central limit theorem criteria).

3.3.2 Significance of the Memorization Gain

Next, we demonstrate the significance of the memorization gain through an example. We again consider a first-order Markov source with alphabet size $k = 256$. We also assume that the source is such that $H_n(\theta)/n = 0.5$ bit per source symbol (byte). In this example, we focus on strictly lossless compression schemes, and hence, $p_e = 0$. In Figure 10, the lower bound on the memorization gain is demonstrated as a function of the sequence length n for different values of the memory size m. As can be seen, significant improvement in the compression may be achieved using memorization. For example, the lower bound on $g_{\mathrm{M}}(32\text{kB}, m, \theta, 0.05, 0)$ is equal to 1.39, 1.92, 2.22, and 2.32, when the context parameter m is 128kB, 512kB, 2MB, and 8MB, respectively. Further, $g_{\mathrm{M}}(512\text{kB}, \infty, \theta, 0.05, 0) = 2.35$. Hence, more than a factor of two improvement is achieved on top of traditional universal compression when network packets of lengths up to 32kB are compression using the memory-assisted compression technique. As demonstrated in Figure 10, the memorization gain for memory of size 8MB is very close to $g_{\mathrm{M}}(n, \infty, \theta, \epsilon, 0)$, and hence, increasing the memory size beyond 8MB does not result in substantial increase of the memorization gain. On the other hand, we further observe that as $n \to \infty$, the memorization gain becomes negligible regardless of the memory size. For example, at $n = 32\text{MB}$ even when $m \to \infty$, we have $g(32\text{MB}, \infty, \theta, 0.05, 0) \approx 1.01$, which is a subtle improvement. This is not surprising as the redundancy that is removed via the memorization technique is $O(\log n/n)$, which becomes negligible as $n \to \infty$.

3.3.3 Analysis of the Required Memory Size

Thus far, we have shown that significant performance improvement is obtained from memory-assisted compression. As also was evident in the example of Section 3.3.2, as the size of the memory increases the performance of the memory-assisted compression is improved. However, there is a certain memory size beyond which the improvement becomes negligible. In this section, we will quantify the required size of memory such that the benefits of the memory-assisted compression apply.

Let $\hat{g}_{\text{M}}(n, \theta, \epsilon, p_e)$ be defined as

$$\hat{g}_{\text{M}} \triangleq 1 + \frac{(1 - p_e)\bar{R}^0_{\text{Ucomp}}(n) + \log\left(\frac{\epsilon}{e}\right) - h(p_e) - p_e H_n(\theta)}{H_n(\theta)}. \tag{70}$$

Then, the following theorem determines the size of the required memory.

Theorem 3.3.6. *Let $m^\star$ be given by*

$$m^\star(\delta, \theta) = \frac{\frac{1-\delta}{\delta}\frac{d}{2}\log e}{\frac{H_n(\theta)}{n}}. \tag{71}$$

Then, for any $m \geq m^\star$, we have

$$g_{\text{M}}(n, m, \theta, \epsilon, p_e) \geq (1 - \delta)\hat{g}_{\text{M}}(n, \theta, \epsilon, p_e).$$

Proof. Using Theorem 3.3.2, we have

$$g_{\text{M}}(n, m, \theta, \epsilon, p_e) \geq \hat{g}_{\text{M}} \frac{H_n(\theta)}{H_n(\theta) + \hat{R}_{\text{M}}(n, m)}. \tag{72}$$

Now, the theorem is proved by considering the following lemma, the proof of which is provided in Appendix A.4.

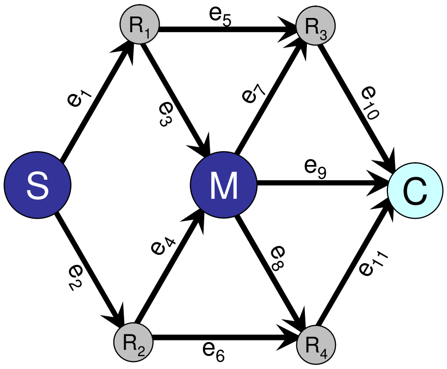

Figure 11: The sample network in case study.

Lemma 3.3.7. *If $m \geq m^{\star}$, we have*

$$\frac{H_n(\theta)}{H_n(\theta) + \hat{R}_{\mathrm{M}}(n, m)} \geq 1 - \delta.$$

□

Theorem 3.3.6 determines the size of the memory that is sufficient for the gain to be at least a fraction $(1 - \delta)$ of the gain obtained as $m \to \infty$. If we again consider the first-order Markov source example of the previous section, with $\delta = 0.01$, we have $m^{\star}(\delta, \theta) \approx 8.9$MB is sufficient for the gain to reach 99% of its maximum confirming our previous observation.

3.4 A Simple Case Study on Network

In this section, we would like to demonstrate as to how the memory-assisted compression gain (over the existing end-to-end compression techniques) is leveraged in terms of the compression of network traffic. The example illustrates as to how we explore the memory element for the purpose of the network traffic compression.

We already demonstrated in Section 2 that the redundancy rate for the universal compression is quite large when the sequence length n is finite. As discussed throughout this work, our proposition in memory-assisted compression is to utilize the memory to achieve a significantly smaller code length. Assume that source S is the CNN server and the packet size is $n = 1$kB. Further, assume that the memory size is 4MB. In Section 2.4, we demonstrated that for this source, the average compression ratio for Ucomp is $\frac{1}{n}El_n(X^n) = 4.42$ bits per byte for this packet size. We further expected that the memorization gain for such packet size be at least $g = 5$. Note that the rest of this discussion is concerned as to how the memorization gain impacts the overall performance in the network.

We now demonstrate the gain of network compression in *bit*×*hop* (BH) for the sample network presented in Figure 11, where M denotes the memory element. Assume that the server S would serve the client C in the network. The intermediate nodes R_i are not capable of memorization. Recall that the network compression gain is measured versus the conventional end-to-end compression. It is also important to recall that the memorization gain g is leveraged on every link that uses memory. For example, in the simple network in Fig 7, the gain of UcompED over Ucomp in terms of flow reduction is only applicable to the links on the S-M path. Hence, for the given memorization gain $g_0 = g_{\text{M}}(n, m, \theta, \epsilon, p_e)$, the number of bits needed to transfer x^n from node S to node M is reduced from $\mathbf{E}l_n(X^n)$ in Ucomp to $\frac{1}{g_0}\mathbf{E}l_n(X^n)$ in UcompED.

To investigate the gain in terms of *bit*×*hop*, we consider the two schemes of UcompED and Ucomp. Let $d(S, C)$ denote the length of the shortest path from S to C, which is clearly $d(S, C) = 3$, e.g., using the path e_1, e_5, e_{10}. Let $\text{BH}(S, C)$ denote the minimum bit-hop cost required to transmit the sequence (of length n) from S to C without any compression mechanism, which is $\text{BH}(S, C) = 24$kbits (which is 1kB×8bits/byte×3). In the case of end-to-end universal compression, i.e., using Ucomp, on the average we need to transmit $\text{BH}_{\text{Ucomp}} = nr_n d(S, C)$ *bit*×*hop* for the transmission of a packet of length n to the client. However, in the case of network compression via memory, i.e., using UcompED, for every information bit on the path from the server to the memory element M, we can leverage the memory, and hence, we require $\frac{1}{n}El_{n,m}(X^n, Y^m)$ bit transmissions per each source symbol that is transmitted to the memory element. Then, the memory element M will decode the received codeword using UcompED decoder and the side information sequence y^m. It will then re-encode the result using Ucomp encoder for the final destination (the client C). In this example, this implies that we require to transmit $\frac{2}{n}\mathbf{E}l_{n,m}(X^n)$ *bit*×*hop* on the average from S to M on links e_1 and e_3 (i.e., $d(S, M) = 2$) for each source symbol. Then, we transmit the message using $\mathbf{E}l_n(X^n)$ *bit*×*hop* per source symbol from M to C on the link e_9. Let $\text{BH}_{\text{UcompED}}$ be the minimum *bit*×*hop* cost for transmitting the sequence (of length n) using the network compression, i.e., $\text{BH}_{\text{UcompED}} = 2\mathbf{E}l_{n,m}(X^n, Y^m) + \mathbf{E}l_n(X^n)$. Let $\mathcal{G}_{\text{BH}}$ be the *bit*×*hop* gain of network compression, defined as $\mathcal{G}_{\text{BH}} = \frac{\text{BH}_{\text{Ucomp}}}{\text{BH}_{\text{UcompED}}}$. Thus, $\mathcal{G}_{\text{BH}} = 2.14$ in this example by substituting $g_0 = 5$. In other words, network compression (using UcompED)achieves more than a factor of 2 saving in *bit*×*hop* over the end-to-end universal compression of the packet (using Ucomp).

3.5 Scaling of the Network-Wide Memorization Gain

In [66, 67], Sardari *et al.* demonstrated that by deploying memory in the network (i.e., enabling some nodes to memorize source sequences), the correlation in the network traffic can be significantly suppressed leading to a much smaller communication cost. In [66, 67], it was assumed that memorization of the previous sequences from the same source provides a fundamental gain g (in the path from the source to a memory node) over and above the performance of the end-to-end universal compression of a new sequence from the same source. Given the link level gain g, the network-wide memorization gain $\mathcal{G}$ on both a random network graph [66] and a power-law network graph [67] were derived when a small fraction of the network nodes are capable of memorization. Further, the scaling of $\mathcal{G}$ with the number of memory elements in the network was obtained. However, [66] and [67] did not explain as to how the link level gain g is computed.

3.6 Technical Analysis: Proof of Theorem 3.3.2

In order to establish the desired lower bound on $g_{\mathrm{M}}(n, m, \theta, \epsilon, p_e)$, we need a lower bound on $R_n(l_n^{p_e}, \theta)$. Further, we require a useful upper bound on $R_n(l_{n,m}^{p_e}, \theta)$.

3.6.1 Lower Bound on the Redundancy of the End-to-End Coding Strategy

In the case of the strictly lossless Ucomp coding strategy, the side information sequence is not utilized for the compression of x^n. It was shown by Clarke and Barron in [24] and later generalized by Atteson in [8] that:

Theorem 3.6.1. *The average minimax redundancy for the strictly lossless Ucomp coding strategy is given by*

$$\bar{R}^0_{Ucomp}(n) = \frac{d}{2}\log\left(\frac{n}{2\pi e}\right) + \log\int_{\theta\in\Lambda^d} |\mathcal{I}(\theta)|^{\frac{1}{2}} d\theta + O\left(\frac{1}{n}\right).$$

In order to prove Theorem 3.3.2, we need a lower bound on the expected redundancy, i.e., $R_n(l_n^{p_e}, \theta)$. To this end, we need a lower bound which was derived in Chapter 2 in Corollary 2.3.6. The result in Corollary 2.3.6 uses $\bar{R}^{p_e}_{\mathrm{Ucomp}}(n)$. By considering the almost lossless Ucomp coding strategy, we demonstrate the following lower bound on $\bar{R}^{p_e}_{\mathrm{Ucomp}}(n)$.

Theorem 3.6.2. *The average minimax redundancy for the lossless Ucomp coding strategy is lower bounded by*

$$\bar{R}^{p_e}_{Ucomp}(n) \geq (1 - p_e)\bar{R}^0_{Ucomp}(n) - h(p_e) - p_e H_n(\theta),$$

where $h(p_e)$ is the binary entropy function defined in (69).

Proof. This is proved in the more general form in Theorem 4.5.1. □

The following lemma helps to combine the results of Corollary 2.3.6 and Theorem 3.6.2 for the desired lower bound.

Lemma 3.6.3. *For any* $\hat{R}_L \in \mathbb{R}$ *such that* $0 < \hat{R}_L \leq \bar{R}^{p_e}_{Ucomp}(n)$*, we have*

$$\mathbf{P}\left[\frac{R_n(l^{p_e}_n, \theta)}{\hat{R}_L} \geq 1 - \delta\right] \geq 1 - e2^{-\delta \hat{R}_L}.$$

Proof. We have

$$\begin{aligned}\mathbf{P}\left[\frac{R_n(l^{p_e}_n, \theta)}{\hat{R}_\mathrm{L}} \geq 1 - \delta\right] &\geq \mathbf{P}\left[\frac{R_n(l^{p_e}_n, \theta)}{\bar{R}^{p_e}_\mathrm{Ucomp}(n)} \geq 1 - \delta\right] && (73)\\ &\geq 1 - e2^{-\delta \bar{R}^{p_e}_\mathrm{Ucomp}(n)} && (74)\\ &\geq 1 - e2^{-\delta \hat{R}_\mathrm{L}}, && (75)\end{aligned}$$

where the inequalities in (73) and (75) are due to $\hat{R}_\mathrm{L} \leq \bar{R}^{p_e}_\mathrm{Ucomp}(n)$, and the inequality in (74) is due to Corrolary 2.3.6. □

3.6.2 Upper Bound on the Redundancy of the Memory-Assisted Compression Strategy

Next, we present the upper bound on for UcompED coding strategy. In the case of strictly lossless UcompED, since a random sequence Y^m is also known to the encoder, the achievable codelength for representing x^n is given by $H(X^n|Y^m)$. Then, the redundancy is given by the following theorem.

Theorem 3.6.4. *The average minimax redundancy for the strictly lossless UcompED coding strategy is*

$$\bar{R}^0_{UcompED}(n, m) = \frac{d}{2}\log\left(1 + \frac{n}{m}\right) + O\left(\frac{1}{n} + \frac{1}{m}\right).$$

Proof. We prove that the right hand side in Theorem 3.6.4 is both an upper bound and a lower bound for $\bar{R}_{\text{UcompED}}(n,m)$. The upper bound is obtained using the KT-estimator [47] along with a Shannon code [74] and the proof follows the analysis of the redundancy of the KT-estimator. In the next lemma, we obtain the lower bound.

Lemma 3.6.5. *The average minimax redundancy of UcompED is lower-bounded by*

$$\bar{R}^0_{UcompED}(n,m) \geq \frac{d}{2}\log\left(1+\frac{n}{m}\right)+O\left(\frac{1}{m}+\frac{1}{n}\right).$$

The proof of Lemma 3.6.5 is provided in Appendix A.5. □

In the case of almost lossless UcompED coding strategy, we have the following upper bound.

Proposition 3.6.6. *The average minimax redundancy for lossless UcompED coding strategy is upper bounded by*

$$\bar{R}^{p_e}_{UcompED}(n,m) \leq \bar{R}^0_{UcompED}(n,m).$$

Proposition 3.6.6 sets the desired upper bound on the average redundancy of the memory-assisted compression.

3.6.3 Main Proof

We are now equipped to present the proof of our main result.

Proof of Theorem 3.3.2. First note that

$$
\begin{aligned}
Q(n,m,\theta,p_e) &= \frac{\mathbf{E}l_n^{p_e}(X^n)}{\mathbf{E}l_{n,m}^{p_e}(X^n)} && (76)\\
&= \frac{H_n(\theta)+R_n(l_n^{p_e},\theta)}{H_n(\theta)+R_n(l_{n,m}^{p_e},\theta)} && (77)\\
&\geq \frac{H_n(\theta)+R_n(l_n^{p_e},\theta)}{H_n(\theta)+\bar{R}^{p_e}_{\text{UcompED}}(n,m)} && (78)\\
&\geq \frac{H_n(\theta)+R_n(l_n^{p_e},\theta)}{H_n(\theta)+\bar{R}^{0}_{\text{UcompED}}(n,m)} && (79)\\
&\triangleq \breve{Q}(n,m,\theta,p_e), && (80)
\end{aligned}
$$

where the inequality in (78) follows from the definition of the average minimax redundancy and the inequality in (79) is due to Proposition 3.6.6. Further, let $\breve{g}(n,m,\theta,\epsilon,p_e)$ be defined as

$$\breve{g}(n,m,\theta,\epsilon,p_e) \triangleq \sup_{z\in\mathbb{R}} \left\{ z : \mathbf{P}\left[\breve{Q}(n,m,\theta,p_e) \geq z\right] \geq 1-\epsilon \right\}. \tag{81}$$

Equation (80) implies

$$\mathbf{P}\left[Q(n,m,\theta,p_e) \geq z\right] \geq \mathbf{P}\left[\breve{Q}(n,m,\theta,p_e) \geq z\right]. \tag{82}$$

Thus, $g_{\text{M}}(n,m,\theta,\epsilon,p_e) \geq \breve{g}(n,m,\theta,\epsilon,p_e)$. We can now apply Corollary 2.3.6 to $R_n(l_n^{p_e},\theta)$ in (79). Then, we can consider Lemma 3.6.3 with $\hat{R}_{\text{L}}$ obtained from the result of Theorem 3.6.2. Then, if δ is chosen such that $\epsilon = e2^{-\delta\hat{R}_L}$, this will provide with a lower bound on $\breve{g}(n,m,\theta,\epsilon,p_e)$, which completes the proof by substituting $\bar{R}^0_{\text{UcompED}}(n,m)$ from Theorem 3.6.4. □

3.7 Conclusion

In this chapter, network compression via network memory was proposed. It was demonstrated that using memorization of the previously seen packets, it is possible to achieve a fundamental improvement over the performance of traditional end-to-end universal compression. The fundamental gain of memorization was defined and a lower bound was derived on the memorization gain. It was concluded that if the intermediate nodes in the network are capable of memorization, significant performance improvement is obtained in the universal compression of network packets. Memory-assisted compression was shown to offer significant improvement above and beyond the conventional end-to-end compression. In summary, this chapter demonstrated that via memorization of the previous packets, the intermediate nodes in the network can learn about the source statistics, which in turn results in noticeable improvement of the performance of the universal compression.

CHAPTER IV

UNIVERSAL COMPRESSION OF DISTRIBUTED SOURCES

A wide variety of applications involve acquiring data from distributed (i.e., spatially separated) sources that cannot communicate with each other, such as, acquiring digital/analog data from sensors [79, 56, 71, 54, 35], the CEO problem [20, 55], delivery of network packets in a content-centric network [44, 40], acquiring data from femtocell wireless networks [23, 22], acquiring data chunks from the cloud storage [7, 36], etc. In all such applications, compression of the data can result in significant performance improvement by saving bandwidth.

The premise of data compression broadly relies on the correlation in the data. For instance, data that are gathered from multiple sensors measuring the same phenomenon (e.g., temperature) are clearly correlated. As another example, when chunks of the same file/content are acquired by a client in a content-centric network, the data chunks are correlated because they are originated from the same file/content. Although there are several formulations for the multi-terminal source coding problem in the literature (cf. [79, 56, 54, 71, 35, 20, 55] and the references therein), there are several emerging scenarios (e.g., the content-centric networks and wireless femtocell networks) that do not fall into the realm of the existing multi-terminal source coding problems (i.e., Slepian-Wolf, Wyner-Ziv, CEO problem, etc). Previous work is mostly concerned about the compression of sequences that bear symbol-by-symbol correlation. On the other hand, the focus of this work is on the universal compression of the data traffic from multiple sources with correlated parameter vectors (which is a different notion of correlated sources that fits the universal compression of the

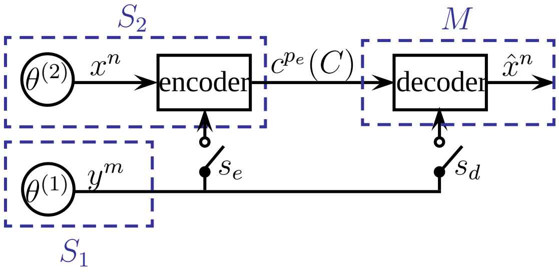

Figure 12: The compression model for universal distributed source coding with correlated parameter vectors.

Internet traffic data).

In this chapter, we introduce and study *universal compression of distributed parametric sources with correlated parameter vectors*. As the most basic case, we assume two parametric sources S_1 and S_2 with unknown parameter vectors $\theta^{(1)}$ and $\theta^{(2)}$, respectively. The source nodes communicate with a destination node M, as shown in Figure 12. We further assume that $\theta^{(1)}$ and $\theta^{(2)}$ are correlated as we will describe the nature of their correlation in detail in Section 4.1. Note that we assume that y^m and x^n are generated as *independent* samples of S_1 and S_2 (given the source parameter vectors are known), respectively. However, when the source parameter vectors are unknown, y^m and x^n are *correlated* with each other through the information they contain about the unknown but correlated source parameter vectors. We wish to leverage this correlation in the encoder of S_2 and the decoder of M in the decoding of x^n by using the side information sequence y^m (from S_1) in order to reduce the average codeword length of x^n.

This problem can also be viewed as universal compression with training data that is available to the encoder and/or the decoder. Thus far, in Chapter 3, we derived the average redundancy that is incurred in the universal compression of the sequence x^n from S_2 when a side information sequence y^m from S_2 (i.e., the same source) is available to *both* the encoder (at S_2) and the decoder (at M). This corresponds to the reduced case of our problem where the correlation between the source parameter vectors is in the form of exact equality, i.e., $\theta^{(2)} = \theta^{(1)}$. Further, we only considered the encoder-decoder side information case, which occurs when the sources S_1 and S_2 are allowed to communicate. However, as we demonstrate in this chapter, the extension to the multiple spatially separated sources, where the training data is not necessarily from the same source model and is only available to the decoder raises a non-trivial set of challenges that are addressed in this work.

The rest of this chapter is organized as follows. In Section 4.1, we describe the source models. In Section 4.2, we present the formal problem setup. In Section 4.3, we provide the definition of the minimum average redundancy as the performance metric for the universal compression of distributed sources. Sections 4.4 and 4.5 give the main results on the average redundancy of strictly lossless and almost lossless codes, respectively. In Section 4.6, we provide some discussion on the significance of our results. Finally Section 4.7 concludes the chapter.

4.1 Source Models: The Nature of Correlation

To proceed, we first review the necessary background. Let $\mathcal{A}$ be a finite alphabet with alphabet size $|\mathcal{A}|$. We define a parametric source by using a d-dimensional parameter vector $\theta = (\theta_1, ..., \theta_d) \in \Lambda$, where d denotes the number of the source parameters and $\Lambda \subset \mathbb{R}^d$ is the space of d-dimensional parameter vectors of interest. Denote μ_θ as the probability measure defined by a parameter vector θ on sequences of length n from the source. In this chapter, we consider two parametric sources S_1 and S_2 with d-dimensional parameter vectors $\theta^{(1)} \in \Lambda$ and $\theta^{(2)} \in \Lambda$, respectively. We further assume that the parameter vectors are a priori unknown to the encoder and the decoder.

Let $x^n = (x_1, ..., x_n) \in \mathcal{A}^n$ be a sequence of length n from the alphabet $\mathcal{A}$. Throughout this chapter we use the notation $X^n \sim \mu_{\theta^{(2)}}$ to denote a random sequence X^n of length n that follows with probability distribution function $\mu_{\theta^{(2)}}$. Similarly, we use the notation $Y^m \sim \mu_{\theta^{(1)}}$ to denote a random sequence of length m that follows $\mu_{\theta^{(1)}}$ (i.e., generated by S_1).

As discussed earlier, the least favorable prior on the space of the parameter vectors is Jeffreys' prior given in (11). Jeffreys' prior is optimal in the sense that the minimum average redundancy is asymptotically achieved by an optimal code when the source parameter vector is assumed to follow Jeffreys' prior [24]. This prior distribution is particularly interesting because it also corresponds to the least favorable prior for the compression performance of the best coding scheme, i.e., it is the capacity achieving distribution.

Next, we present our model for the nature of the correlation between the parameter vectors of the sources S_1 and S_2. In this model, as we shall see, the correlation between the two sources is controlled using the single parameter $\mathcal{T}$. Assume that $\theta^{(1)}$ follows Jeffreys's prior, i.e., $\theta^{(1)} \sim p_J$ or $p_{\theta^{(1)}}(\theta^{(1)}) = p_J(\theta^{(1)})$. Let $Z^{2\mathcal{T}}$ be a random sequence of length $2\mathcal{T}$ that follows $\mu_{\theta^{(1)}}$. We further assume that given $Z^{2\mathcal{T}}$, the parameter vectors $\theta^{(1)}$ and $\theta^{(2)}$ are independent and identically distributed, i.e., $p_{\theta^{(2)}|z^{2\mathcal{T}}}(\cdot) = p_{\theta^{(1)}|z^{2\mathcal{T}}}(\cdot)$. Then, the conditional distribution of $\theta^{(2)}$ given $\theta^{(1)}$, i.e., $p^{\mathcal{T}}_{\theta^{(2)}|\theta^{(1)}}(\cdot)$, is given by the following:

$$p^{\mathcal{T}}_{\theta^{(2)}|\theta^{(1)}}(\theta^{(2)}|\theta^{(1)}) = \sum_{z^{2\mathcal{T}} \in \mathcal{A}^{2\mathcal{T}}} p(\theta^{(2)}, z^{2\mathcal{T}}|\theta^{(1)}) \tag{83}$$

$$= \sum_{z^{2\mathcal{T}}} p_{\theta^{(1)}|z^{2\mathcal{T}}}(\theta^{(2)}|z^{2\mathcal{T}}) \mu_{\theta^{(1)}}(z^{2\mathcal{T}}) \tag{84}$$

$$= p_J(\theta^{(2)}) \sum_{z^{2\mathcal{T}}} \left(\frac{\mu_{\theta^{(1)}}(z^{2\mathcal{T}})\mu_{\theta^{(2)}}(z^{2\mathcal{T}})}{\int_{\lambda\in\Lambda} \mu_\lambda(z^{2\mathcal{T}}) p_J(\lambda) d\lambda} \right). \tag{85}$$

This conditional distribution results in the joint distribution defined in Definition 4.1.1.

Definition 4.1.1. *Let $\theta^{(1)}$ and $\theta^{(2)}$ denote the parameter vectors associated with S_1 and S_2, respectively. Then, S_1 and S_2 are said to be correlated with degree $\mathcal{T}$ if $(\theta^{(1)}, \theta^{(2)})$ have the following joint probability distribution:*

$$p^{\mathcal{T}}_{\theta^{(1)},\theta^{(2)}}(\theta^{(1)}, \theta^{(2)}) \triangleq p_J(\theta^{(1)})p_J(\theta^{(2)}) \sum_{z^{2\mathcal{T}} \in \mathcal{A}^{2\mathcal{T}}} \left(\frac{\mu_{\theta^{(1)}}(z^{2\mathcal{T}})\mu_{\theta^{(2)}}(z^{2\mathcal{T}})}{\int_{\lambda\in\Lambda} \mu_\lambda(z^{2\mathcal{T}}) p_J(\lambda) d\lambda} \right).[1]$$

[1] *Please note that we use the superscript $\mathcal{T}$ to stress that the conditional distribution is defined as a function of the parameter $\mathcal{T}$.*

In the following, we state some of the nice properties of this correlation model.

Lemma 4.1.2. *The marginal distributions of $\theta^{(1)}$ and $\theta^{(2)}$ are Jeffreys' prior, i.e., $p_{\theta^{(1)}}(\theta^{(1)}) = p_J(\theta^{(1)})$ and $p_{\theta^{(2)}}(\theta^{(2)}) = p_J(\theta^{(2)})$.*

Proof.

$$p_{\theta^{(2)}}(\theta^{(2)}) = \int_{\theta^{(1)}\in\Lambda} p^{\mathcal{T}}_{\theta^{(1)},\theta^{(2)}}(\theta^{(1)},\theta^{(2)})d\theta^{(1)} \tag{86}$$

$$= p_J(\theta^{(2)})\sum_{z^{2\mathcal{T}}}\left(\frac{\mu_{\theta^{(2)}}(z^{2\mathcal{T}})\int_{\theta^{(1)}\in\Lambda}\mu_{\theta^{(1)}}(z^{2\mathcal{T}})p_J(\theta^{(1)})d\theta^{(1)}}{\int_{\lambda\in\Lambda}\mu_\lambda(z^{2\mathcal{T}})p_J(\lambda)d\lambda}\right) \tag{87}$$

$$= p_J(\theta^{(2)})\sum_{z^{2\mathcal{T}}}\mu_{\theta^{(2)}}(z^{2\mathcal{T}}) = p_J(\theta^{(2)}). \tag{88}$$

The claim for $p_{\theta^{(1)}}(\theta^{(1)})$ is proved similarly and is omitted for brevity. □

Lemma 4.1.3. *The conditional distribution of $\theta^{(1)}$ given $\theta^{(2)}$ is symmetric, i.e.,*

$$p^{\mathcal{T}}_{\theta^{(2)}|\theta^{(1)}}(y|x) = p^{\mathcal{T}}_{\theta^{(1)}|\theta^{(2)}}(y|x).$$

Proof. The proof is competed by using Bayes' rule and Lemma 4.1.2. □

According to Lemma 4.1.3, the conditional distribution $p^{\mathcal{T}}_{\theta^{(2)}|\theta^{(1)}}$ is symmetric. Thus, there is no distinction between S_1 and S_2. In other words, for any given $\mathcal{T}$, the problem of the universal compression of a sequence x^n from S_2 given a side information sequence from S_1 becomes equivalent to the problem of the compression of a sequence x^n from S_1 given a side information sequence from S_2.

Lemma 4.1.4. *If $\mathcal{T} = 0$, then $\theta^{(2)}$ and $\theta^{(1)}$ are independent, i.e.,*

$$p^{0}_{\theta^{(2)}|\theta^{(1)}}(\theta^{(2)}|\theta^{(1)}) = p_J(\theta^{(2)}).$$

Proof. We have

$$p^{0}_{\theta^{(2)}|\theta^{(1)}}(\theta^{(2)}|\theta^{(1)}) = \frac{p_J(\theta^{(2)})}{\int_{\lambda\in\Lambda} p_J(\lambda)d\lambda} = p_J(\theta^{(2)}), \tag{89}$$

which completes the proof. □

According to Lemma 4.1.4, we can make $\theta^{(1)}$ and $\theta^{(2)}$ independent by setting $\mathcal{T} = 0$. In this case, both $\theta^{(1)}$ and $\theta^{(2)}$ will follow Jeffreys' prior, i.e., identically distributed.

Lemma 4.1.5. *The parameter vector $\theta^{(2)}$ converges to $\theta^{(1)}$ in probability as $\mathcal{T} \to \infty$.*

Proof. See Appendix A.6. □

According to Lemma 4.1.5, when $\mathcal{T} \to \infty$, we have $\theta^{(2)} \to \theta^{(1)}$ in probability, which reduces to the universal compression of distributed identical sources problem studied in [17].

Remark: The degree of correlation between the two parameter vectors $\theta^{(1)}$ and $\theta^{(2)}$ is determined via the parameter $\mathcal{T}$. This degree of correlation varies from independence of the two parameter vectors at $\mathcal{T} = 0$ all the way to their equality when $\mathcal{T} \to \infty$.

4.2 Problem Setup

In this section, we present the basic setup of the problem. As shown in Figure 12, the setup is comprised of two sources S_1 and S_2. Let y^m and x^n denote two sequences (samples) of lengths m and n, respectively, that are generated by S_1 and S_2, i.e., samples from $\theta^{(1)}$ and $\theta^{(2)}$. Here, we only consider the case where $m = \omega(n)$,[2] i.e., the length of the side information sequence m is growing with a larger rate than the sequence length n. We consider four coding strategies (according to the orientation

[2] $f(n) = \omega(n)$ iff $\lim_{n\to\infty} \frac{n}{f(n)} = 0$.

of the switches s_e and s_d in Figure 12) for the compression of x^n from S_2 provided that the sequence y^m from S_1 is available to the encoder/decoder or not.[3]

- Ucomp (Universal compression without side information), where the switches s_e and s_d in Figure 12 are both *open*. In this case, the encoder input space is given by $\mathcal{C} = \mathcal{A}^n \times \mathbb{Z}^*$. We let $C = (x^n, m)$ denote the input to the encoder. The decoder input space is denoted by $\mathcal{D} = \{0,1\}^* \times \mathbb{Z}^*$ and we let $D = (c(C), m)$ denote the input to the decoder.

- UcompE (Universal compression with encoder side information), where the switch s_e in Figure 12 is *closed* but the switch s_d is *open*. In this case, the encoder input space is given by $\mathcal{C}^E = \mathcal{A}^n \times \mathcal{A}^m$. We let $C^E = (x^n, y^m)$ denote the input to the encoder. The decoder input space is denoted by $\mathcal{D}^E = \{0,1\}^* \times \mathbb{Z}^*$ and we let $D^E = (c(C^E), m)$ denote the input to the decoder.

- UcompD (Universal compression with decoder side information), where the switch s_e in Figure 12 is *open* but the switch s_d is *closed*. In this case, the encoder input space is given by $\mathcal{C}^D = \mathcal{A}^n \times \mathbb{Z}^*$. We let $C^D = (x^n, m)$ denote the input to the encoder. The decoder input space is denoted by $\mathcal{D}^D = \{0,1\}^* \times \mathcal{A}^m$ and we let $D^D = (c(C^D), y^m)$ denote the input to the decoder.

- UcompED (Universal compression with encoder-decoder side information), where the switches s_e and s_d in Figure 12 are both *closed*. In this case, the encoder input space is given by $\mathcal{C}^{ED} = \mathcal{A}^n \times \mathcal{A}^m$. We let $C^{ED} = (x^n, y^m)$ denote the input to the encoder. The decoder input space is denoted by $\mathcal{D}^{ED} = \{0,1\}^* \times \mathcal{A}^m$ and we let $D^{ED} = (c(C^{ED}), y^m)$ denote the input to the decoder.

Please note that, from the viewpoint of applications, the interesting coding strategy in this study is UcompD, where the side information sequence from S_1 is available at

[3] In this chapter, we assume that m and n are a priori known to both the encoder and the decoder.

the decoder while it is not known to the encoder. The Ucomp coding is the benchmark for the achievable universal compression when no side information is present. Further, UcompED is the benchmark in the evaluation of UcompD but it may not be practically useful since it requires the sequence y^m from S_1 to be available at the encoder of S_2, i.e., coordination between the two encoders. Finally, UcompE is presented for the sake of completeness and as will be revealed, UcompE provides no significant improvements over Ucomp.

In this chapter, we focus on the family of fixed-to-variable length codes that map an n-vector to a variable-length binary sequence [81]. We only consider codes that are uniquely decodable, i.e., satisfy Kraft inequality.

Definition 4.2.1. *The code $c : \mathcal{C} \to \{0,1\}^*$ is called strictly lossless (also called zero-error) if there exists a reverse mapping $d : \mathcal{D} \to \mathcal{A}^n$ such that*

$$\forall x^n \in \mathcal{A}^n : \quad d(D) = x^n.$$

Most of the practical data compression schemes are examples of strictly lossless codes, namely, the arithmetic coding [49], Huffman [38], Lempel-Ziv [94, 95], and context-tree-weighting (CTW) algorithm [88]. Please note that based on the orientation of the switches in Figure 12 and the input and output spaces, it is straightforward to extend the definition of strictly lossless codes to UcompE, UcompD, and UcompED.

On the other hand, due to the distributed nature of the sources, we are also concerned with the slightly weaker notion of almost lossless source coding:

Definition 4.2.2. *The code $c^{p_e} : \mathcal{C} \to \{0,1\}^*$ is called almost lossless with permissible error probability $p_e(n) = o(1)$, if there exists a reverse mapping $d^{p_e} : \mathcal{D} \to \mathcal{A}^n$ such that*

$$\mathbf{E}\{\mathbf{1}_e\} \leq p_e(n),$$

where $\mathbf{1}_e(x^n, \hat{x}^n)$ *denotes the error indicator function and* $\hat{x}^n = d^{p_e}(D)$*, i.e,*

$$\mathbf{1}_e(x^n, \hat{x}^n) \triangleq \begin{cases} 1 & \hat{x}^n \neq x^n, \\ 0 & \textit{otherwise.} \end{cases}$$

This definition also extends to UcompE, UcompD, and UcompED. Please also note that in this definition we have slightly abused the notation to denote D as the input to the decoder when the encoder is almost lossless as well as strictly lossless.

The almost lossless codes allow a non-zero error probability $p_e(n)$ for any finite n while they are *almost surely* asymptotically error free. Note that almost lossless codes with $p_e(n) = 0$ are indeed strictly lossless codes. Thus, we also use the notation c^0 to denote a strictly lossless code. The proof of the Slepian-Wolf Theorem [79] uses almost lossless codes. Further, all of the practical implementations of Slepian-Wolf source coding are based on almost lossless codes (cf. [56] and [71]). We stress that the nature of the almost lossless source coding is different from that incurred by the lossy source coding (i.e., the rate-distortion theory). In the lossy source coding, a code is designed to asymptotically achieve a given distortion level as the length of the sequence grows to infinity. Therefore, since the almost lossless coding asymptotically achieves a zero-distortion, it coincides with the special case of zero-distortion in the rate-distortion curve.

4.3 Minimum Average Redundancy

Let $l^{p_e} : \mathcal{C} \to \mathbb{R}$ denote the universal length function for Ucomp coding with permissible error probability p_e.[4] Denote L^{p_e} as the space of almost lossless universal length functions. Let $R(l^{p_e}, \theta)$ be the expected redundancy of the code with length function $l^{p_e}(\cdot)$, defined in (63). Define $\bar{R}_n^{p_e}$ as the minimum average redundancy of

[4] Note that we have ignored the integer constraint on the length functions in our work, which will result in a negligible $O(1)$ redundancy that is exactly analyzed in [30, 81].

Ucomp coding when the parameter vector θ is chosen using Jeffreys' prior, i.e.,

$$\bar{R}^{p_e} = \min_{l^{p_e}} \int_{\theta \in \Lambda} R(l^{p_e}, \theta) p_J(\theta) d\theta. \tag{90}$$

It is evident that $\bar{R}^0$ is the average maximin redundancy for the strictly lossless compression since θ is known to follow the least favorable Jeffreys' prior (i.e., the capacity achieving prior). It is straightforward to extend the definitions of the length function and the minimum average redundancy to UcompE, UcompD, and UcompED coding strategies that are denoted by $\bar{R}_E^{p_e}$, $\bar{R}_D^{p_e}$, and $\bar{R}_{ED}^{p_e}$, respectively. Furthermore, $\bar{R}^0$ and $\bar{R}_{ED}^0$ are also equal to the average minimax redundancy [34, 52]. Please note that the average redundancy defined here is the fundamental overhead in the compression for the case $p_e = 0$, i.e., for the strictly lossless codes. For the general almost lossless codes, $H_n(\theta)$ no longer serves as the length of the optimal code in the compression of a sequence of length n. On the other hand, the average minimum redundancy defined in (90) can still be employed in the comparison between different strategies for a given p_e.

The following is a trivial statement about the performance of the almost lossless coding strategy.

Lemma 4.3.1. *If $p_e^2 \geq p_e^1 \geq 0$, then $\bar{R}^{p_e^2} \leq \bar{R}^{p_e^1}$.*

Proof. Let $\check{l}^{p_e^1}$ denote code that achieves the permissible error probability p_e^1. By definition $\check{l}^{p_e^1}$ also achieves the permissible error probability p_e^2, which completes the proof. □

The same property holds true for $\bar{R}_E^{p_e}$, $\bar{R}_D^{p_e}$, and $\bar{R}_{ED}^{p_e}$. The following is obtained as a simple corollary to Lemma 4.3.1.

Lemma 4.3.2. *$\forall p_e \geq 0$, we have $\bar{R}^{p_e} \leq \bar{R}^0$.*

In other words, the strictly lossless codes incur a larger redundancy among all almost lossless codes. The following intuitive inequalities demonstrate that the redundancy decreases with the availability of the side information.

Lemma 4.3.3. *$\forall p_e \geq 0$, the following set of inequalities hold:*

$$\begin{cases} \bar{R}^{p_e}_{ED} \leq \bar{R}^{p_e}_{E} \leq \bar{R}^{p_e} \\ \bar{R}^{p_e}_{ED} \leq \bar{R}^{p_e}_{D} \leq \bar{R}^{p_e}. \end{cases} \tag{91}$$

Proof. Let $\check{l}^{p_e} \in L^{p_e}$ denote the optimal code with permissible error probability p_e. Then, it is straightforward to see that $\check{l}^{p_e} \in L^{p_e}_E$ (i.e., $\check{l}^{p_e}$ is a code with permissible error probability p_e and encoder side information) since the encoder can choose not to use the side information sequence y^m in the coding. Likewise, if $\check{l}^{p_e}_E \in L^{p_e}_E$ is the optimal code with permissible error probability p_e and encoder side information. We have $\check{l}^{p_e}_E \in L^{p_e}_{ED}$ as it is a candidate code for the encoder-decoder side information case when the coding system is only a function of the side information sequence length at the decoder and not the side information sequence itself. This completes the proof of the first set of inequalities. The proof for the second set of the inequalities is similar and is omitted for brevity. □

Before we delve into the main results of this chapter, we present another result that will be useful in characterizing the redundancy in later sections.

Lemma 4.3.4. *If $\mathcal{T} = 0$, we have*

$$\bar{R}^{p_e}_{ED} = \bar{R}^{p_e}_{D} = \bar{R}^{p_e}_{E} = \bar{R}^{p_e}.$$

Proof. It suffices to show that $\bar{R}^{p_e}_{ED} = \bar{R}^{p_e}$. Then, by Lemma 4.3.3, the rest follows. As pointed out in Lemma 4.1.4, at case $\mathcal{T} = 0$ $\theta^{(1)}$ and $\theta^{(2)}$ both follow Jeffreys' prior and are *independent*. Then, it is easy to see that in general

$$\begin{aligned} & p^{\mathcal{T}}(x^n|y^m) \\ & = \int\limits_{(\theta^{(1)},\theta^{(2)})\in\Lambda^2} \mu_{\theta^{(2)}}(x^n) p^{\mathcal{T}}_{\theta^{(2)}|\theta^{(1)}}(\theta^{(2)}|\theta^{(1)}) p(\theta^{(1)}|y^m) d\theta^{(1)} d\theta^{(2)}. \end{aligned} \tag{92}$$

Then, since when $\mathcal{T} = 0$, $\theta^{(1)}$ and $\theta^{(2)}$ are independent, we have $p^0_{\theta^{(2)}|\theta^{(1)}}(\theta^{(2)}|\theta^{(1)}) = p_J(\theta^{(2)})$. Thus, we can further simplify (92) to get

$$p^0(x^n|y^m) = p(x^n), \tag{93}$$

which implies that x^n and y^m are independent. Therefore, $H(X^n|Y^m) = H(X^n)$ which implies that the knowledge of the side information sequence Y^m does not decrease the number of bits needed to describe X^n. □

According to Lemma 4.3.4, there is no benefit provided by the side information when the two parameter vectors of the sources S_1 and S_2 are independent. This is not surprising as when $\theta^{(1)}$ and $\theta^{(2)}$ are independent, then X^n (produced by S_1) and Y^m (produced by S_2) are also independent. Thus, the knowledge of y^m does not affect the distribution of x^n. Hence, y^m cannot be used toward the reduction of the codeword length for x^n. Hence, In the case $\mathcal{T} = 0$, the performance of all strategies is equivalent to $\bar{R}^{p_e}$, which will be characterized in later sections.

Lemma 4.3.5. *If $\mathcal{T} = o(n)$, we have*

$$\bar{R}^{p_e}_{ED} = \bar{R}^{p_e} + O(1).$$

Proof. In this case, the nature of the correlation is pretty similar to the observation of a sequence of length $O(\mathcal{T}) = o(n)$ from the source. This does not provide reduction in the average codeword length as it will be dominated by learning the statistics for the compression of a sequence of length n, which requires an average of $\frac{d}{2}\log n + O(1)$ bits. □

Lemma 4.3.5 states that when $\mathcal{T}$ is not sufficiently large (i.e., when the two parameter vectors are not sufficiently correlated), knowledge of a sequence from one would not provide much reduction in the compression of a sequence from the other.

4.4 Performance of Strictly Lossless Codes

In this section, we present our main results on the minimum average redundancy for strictly lossless codes. As previously discussed, we only consider the case where $m = \omega(n)$, i.e., when the size of the side information sequence is sufficiently large. In other words, our focus is not on the transient period where the memory is populated with data traffic. Instead, we would like to analyze how much performance improvement is obtained when a sufficiently large side information sequence is used in the compression of a new sequence.

In the case of Ucomp, the side information sequence is not utilized at the encoder/decoder for the compression of x^n, and hence, the minimum number of bits required to represent x^n is $H(X^n)$. This coincides with the avearge minimax redundancy characterized in 2.3.4. Hence,

$$\bar{R}^0 = \frac{d}{2} \log \left(\frac{n}{2\pi e} \right) + \log \int_{\lambda \in \Lambda} |\mathcal{I}(\lambda)|^{\frac{1}{2}} d\lambda + o(1).^5$$

Next, we confine ourselves to UcompE strategy and establish that the side information provided by y^m only at the encoder does not provide any benefit on the strictly lossless universal compression of the sequence x^n.

Proposition 4.4.1. *The minimum average redundancy for strictly lossless UcompE coding is*

$$\bar{R}^0_E = \bar{R}^0.$$

Proof. In the case of UcompE coding, since the side information sequence y^m is not available to the decoder, then the minimum number of average bits required at the decoder to describe the random sequence X^n is indeed $H(X^n)$. On the other hand,

[5] $f(n) = o(g(n))$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

it is straightforward to see that $H(X^n) = H_n(\theta^{(2)}) + I(X^n; \theta^{(2)})$. Further, it is clear that

$$I(X^n; \theta^{(2)}) = \bar{R}^0. \tag{94}$$

by the redundancy-capacity theorem (cf. [52]). □

Considering the UcompD strategy, we establish a result that the side information provided by y^m at the decoder does not provide any performance improvement in the strictly lossless universal compression of the sequence x^n.

Proposition 4.4.2. *The minimum average redundancy for strictly lossless UcompD coding is*

$$\bar{R}^0_D = \bar{R}^0.$$

Proof. Since the two sources $\mu_{\theta^{(1)}}$ and $\mu_{\theta^{(2)}}$ are assumed to be from the d-dimensional parametric sources, in particular, they are also *ergodic*. In other words, any pair $(x^n, y^m) \in \mathcal{A}^n \times \mathcal{A}^m$ occurs with non-zero probability and the support set of (x^n, y^m) is equal to the entire $\mathcal{A}^n \times \mathcal{A}^m$. Therefore, the knowledge of the side information sequence y^m at the decoder does not rule out any possibilities for x^n at the decoder, and hence, the probability distribution of x^n remains unchanged (equal to the prior distribution) after y^m has been observed. Proposition 4.4.2 is then completed by using the known results of strictly lossless compression (cf. [3] and the references therein). □

Finally, we present our main result on the strictly lossless UcompED coding. In this case, since a side information sequence y^m is known to both the encoder and the decoder, the achievable codeword length for representing x^n is given by $H(X^n|Y^m)$. Hence, the redundancy can be shown to be obtained from the following theorem.

Theorem 4.4.3. *For strictly lossless UcompED coding,* **(a)** *in the case of* $\mathcal{T} = o(n)$, *we have*

$$\bar{R}^0_{ED} = \bar{R}^0 + O(1),$$

and **(b)** *in the case of* $\mathcal{T} = \omega(n)$, *we have*

$$\bar{R}^0_{ED} = \hat{R}(n, m, \mathcal{T}) + o(1),$$

where $\hat{R}(n, m, \mathcal{T})$ *is defined as*

$$\hat{R}(n, m, \mathcal{T}) = \frac{d}{2} \log \left(1 + \frac{n}{m} + \frac{n}{\mathcal{T}} \right). \tag{95}$$

Proof. Part (a) is obtained from Lemma 4.3.5. To prove Part (b), let $X^n \sim \mu_{\theta^{(2)}}$ and $Y^m \sim \mu_{\theta^{(1)}}$. Further, since the encoder also has access to y^m and for $m = \omega(n)$, we can use $\hat{\theta}(y^m)$ (i.e., the estimate of $\theta^{(2)}$ via y^m) for the compression of the sequence x^n. In this case, $\bar{R}_{ED}$ can be written as

$$\begin{aligned}\bar{R}_{ED} &= \mathbf{E} \left\{ D(\mu_{\theta^{(2)}} || \mu_{\hat{\theta}(Y^m)}) \right\} \\ &\eqsim \frac{1}{2} \mathbf{E} \left\{ (\theta^{(2)} - \hat{\theta}(Y^m))^T I(\theta^{(2)}) (\theta^{(2)} - \hat{\theta}(Y^m)) \right\}, ^{6} \qquad (96)\end{aligned}$$

where (96) follows from the following:

$$\begin{aligned}D(\mu_\phi || \mu_\lambda) = &\frac{1}{2}(\lambda - \phi)^T \mathcal{I}(\phi)(\lambda - \phi) \\ &+ o(||\lambda - \phi||^2). \qquad (97)\end{aligned}$$

Please also note that

$$\theta^{(2)} - \hat{\theta}(Y^m) = (\theta^{(2)} - \theta^{(1)}) + (\theta^{(1)} - \hat{\theta}(Y^m)). \tag{98}$$

Further, by definition $\theta^{(2)} - \theta^{(1)}$ and $\hat{\theta}(Y^m) - \theta^{(1)}$ are independent of each other. Hence, we have

$$\frac{1}{2} \mathbf{E} \left\{ (\theta^{(2)} - \hat{\theta}(Y^m))^T I(\theta^{(2)}) (\theta^{(2)} - \hat{\theta}(Y^m)) \right\}$$

[6] $f(n) \eqsim g(n)$ iff $f(n) - g(n) = o(f(n))$.

$$= \frac{d}{2}\left(\frac{n}{\mathcal{T}} + \frac{n}{m}\right)\log e \approx \frac{d}{2}\log\left(1 + \frac{n}{\mathcal{T}} + \frac{n}{m}\right). \tag{99}$$

□

4.5 Performance of Almost Lossless Codes

In this section, we evaluate the performance of each of the different coding schemes introduced in Section 4.2 for the universal compression of distributed parametric sources using their corresponding minimum average redundancy for almost lossless codes.

In the case of almost lossless coding, we have the following result on the performance of Ucomp coding.

Theorem 4.5.1. *The minimum average redundancy for almost lossless Ucomp coding is bounded by*

$$(1 - p_e)\bar{R}^0 - h(p_e) - p_e H_n(\theta^{(2)}) \;\leq\; \bar{R}^{p_e} \;\leq\; \bar{R}^0,$$

where

$$h(p_e) = p_e \log\left(\frac{1}{p_e}\right) + (1 - p_e)\log\left(\frac{1}{1 - p_e}\right). \tag{100}$$

Proof. The upper limit is obvious from Lemma 4.3.2. In order to prove the lower limit, we consider $H(X^n, \hat{X}^n, \mathbf{1}_e)$, where $\hat{X}^n$ is the decoded vector, i.e., $\hat{X}^n = d(D)$. . Note that $\mathbf{1}_e(X^n, \hat{X}^n)$ is a deterministic function of X^n and $\hat{X}^n$ and hence

$$H(X^n, \hat{X}^n, \mathbf{1}_e) = H(X^n) + H(\hat{X}^n | X^n). \tag{101}$$

On the other hand, we can also use the chain rule in a different order to arrive at the following.

$$\begin{aligned} H(X^n, \hat{X}^n, \mathbf{1}_e) = H(\hat{X}^n) + H(\mathbf{1}_e | \hat{X}^n) \\ + H(X^n | \mathbf{1}_e, \hat{X}^n). \end{aligned} \tag{102}$$

Hence,

$$
\begin{aligned}
H(\hat{X}^n) &= H(X^n) + H(\hat{X}^n|X^n) - H(\mathbf{1}_e|\hat{X}^n) - H(X^n|\mathbf{1}_e, \hat{X}^n) \\
&\geq H(X^n) - h(p_e) - H(X^n|\mathbf{1}_e, \hat{X}^n) \qquad (103) \\
&\geq H(X^n) - h(p_e) - p_e H(X^n), \qquad (104)
\end{aligned}
$$

where the inequality in (103) is due to the facts that $H(\hat{X}^n|X^n) \geq 0$ and $H(\mathbf{1}_e|\hat{X}^n) \leq H(\mathbf{1}_e) \leq h(p_e)$ and the inequality in (104) is due to Lemma 4.5.2.

Lemma 4.5.2. *$H(X^n|\mathbf{1}_e, \hat{X}^n) \leq p_e H(X^n)$.*

The proof of Lemma 4.5.2 is provided in Appendix A.7. The proof of the theorem is completed by noting that $H(X^n) = H_n(\theta) + \bar{R}^0$. □

Next, we consider almost lossless UcompE coding. In this case, similar to the strictly lossless case, we prove that the side information provided by y^m at the decoder does not provide any benefit on the almost lossless universal compression of the sequence x^n.

Proposition 4.5.3. *The minimum average redundancy for almost lossless UcompE coding is*

$$\bar{R}_E^{p_e} = \bar{R}^{p_e}.$$

Proof. The proof follows arguments similar to the proof of Proposition 4.4.1 by noting that the decoder needs to describe the random sequence X^n with a permissible error probability p_e. □

In the case of almost lossless UcompD coding, the permissible error probability p_e results in further reduction in the average codeword length relative to Ucomp as given by the following theorem.

Theorem 4.5.4. *For the minimum average redundancy for lossless UcompD coding* **(a)** *in the case of $\mathcal{T} = o(n)$, $\bar{R}_D^{p_e}$ is given by*

$$\bar{R}_D^{p_e} = \bar{R}^0 + O(1),$$

and **(b)** *in the case of $\mathcal{T} = \omega(n)$, $\bar{R}_D^{p_e}$ is bounded by*

$$\bar{R}_{ED}^{p_e} \;\leq\; \bar{R}_D^{p_e} \;\leq\; \bar{R}_{ED}^{0} + \mathcal{F}(d, p_e) + o(1),$$

where $\mathcal{F}(d, p_e)$ is the penalty due to the absence of the side information at the encoder, which is given by

$$\mathcal{F}(d, p_e) = \frac{d}{2} \log \left(1 + \frac{2}{d \log e} \log \frac{4}{p_e} \right). \tag{105}$$

Proof. Part (a) is trivial by combining Lemmas 4.3.3 and 4.3.5. Regarding Part (b), the lower limit is previously proved as part of Lemma 4.3.3. In order to obtain the upper limit on the average redundancy of the almost lossless UcompD coding, we provide a constructive optimal coding strategy at the encoder such that the sequence can be decoded with error smaller than the permissible error probability p_e at the decoder, and obtain its achievable minimum average redundancy. Prior to introducing the coding scheme, we need to establish some results that will be used in the construction of the coding scheme.

Definition 4.5.5. *Let $\Delta_d(\phi, \delta) \subset \Lambda$ be the d-dimensional ellipsoid with radius δ around the source parameter vector ϕ defined as the following:*

$$\Delta_d(\phi, \delta) \triangleq \left\{ \lambda \in \Lambda |\; (\lambda - \phi)^T \mathcal{I}(\phi)(\lambda - \phi) < \delta \right\}, \tag{106}$$

where $\mathcal{I}(\phi)$ is the Fisher information matrix.

Let $\hat{\theta}(x^n)$ denote the maximum likelihood estimate of the parameter vector given the observed sequence x^n, i.e.,

$$\hat{\theta}(x^n) \triangleq \arg \sup_{\lambda \in \Lambda} \mu_\lambda(x^n). \tag{107}$$

Next, we state the following result on the probability of the event that the maximum-likelihood estimate is far from the true parameter vector.

Lemma 4.5.6. *Let $X^n \sim \mu_\theta$ be a sequence of length n from alphabet $\mathcal{A}$ that follows the probability measure μ_θ. Further, let $\hat{\theta}(X^n)$ be the maximum likelihood estimate defined in* (107)*. Then, we have*

$$\mathbf{P}\left[\hat{\theta}(X^n) \notin \Delta_d(\theta, \delta)\right] \eqsim Q_d\left(\delta\frac{n}{2}\right), \tag{108}$$

where

$$Q_d(x) \triangleq \begin{cases} \frac{1}{\Gamma(\frac{d}{2})} x^{d-2} e^{-x^2} & x \geq 0 \\ 0 & x < 0 \end{cases}. \tag{109}$$

The proof of Lemma 4.5.6 is provided in Appendix A.8. Now, let $Q_d^{-1}(\cdot)$ denote the inverse of the $Q_d(\cdot)$. Further, let p_e denote a permissible error probability on a sequence of length n. Then, we have

$$\delta = \frac{2}{n} Q_d^{-1}(p_e). \tag{110}$$

In other words, if we only consider the random sequences X^n, such that $\hat{\theta}(X^n) \in \Delta_d(\theta, \delta)$, then with probability at least $1 - p_e$ any X^n is covered in this set. In the following, we will extend this argument so that we will obtain an ellipsoid around $\hat{\theta}(Y^m)$ (instead of θ) such that X^n is covered with probability at least $1 - p_e$.

The following is an upper limit on the probability of the event that $\theta^{(2)}$ is far from $\theta^{(1)}$.

Lemma 4.5.7. *Let $(\theta^{(1)}, \theta^{(2)}) \sim p^{\mathcal{T}}_{\theta^{(1)},\theta^{(2)}}$ be a pair of random parameter vectors in the space Λ^2 chosen according to the distribution $p^{\mathcal{T}}_{\theta^{(1)},\theta^{(2)}}$ defined in* (86)*. Then, we have*

$$\mathbf{P}\left[\theta^{(2)} \notin \Delta_d(\theta^{(1)}, \delta)\right] \lesssim 2Q_d\left(\delta\frac{\mathcal{T}}{2}\right).^{7} \tag{111}$$

where Q_d is defined in (109)*.*

[7] *$f(n) \lesssim g(n)$ iff $\forall \epsilon > 0\ \exists N_0$ s.t. $\frac{f(n)}{g(n)} < 1 + \epsilon$ for $n > N_0$.*

The proof of Lemma 4.5.7 is provided in Appendix A.9.

Now, we are equipped to present our result on the probability of the event that $\hat{\theta}(X^n)$ is far from $\hat{\theta}(Y^m)$, which is the following:

Lemma 4.5.8. *Let $Y^m \sim \mu_{\theta^{(1)}}$ and $X^n \sim \mu_{\theta^{(2)}}$ denote two independent (given $\theta^{(1)}$ and $\theta^{(2)}$) random samples from S_1 and S_2, respectively. Then, we have the following upper limit*

$$\mathbf{P}\left[\hat{\theta}(X^n) \notin \Delta_d\left(\hat{\theta}(Y^m), \delta\right)\right] \lesssim 4Q_d\left(\frac{\delta}{4}\left(\frac{1}{n} + \frac{1}{\mathcal{T}} + \frac{1}{m}\right)\right),$$

where $m^\star \triangleq \frac{m\mathcal{T}}{m+\mathcal{T}}$ is the effective size of the side information sequence.

The proof of Lemma 4.5.8 is provided in Appendix A.10. Now, we can set

$$p_e = 4Q_d\left(\frac{\delta}{4}\left(\frac{1}{n} + \frac{1}{\mathcal{T}} + \frac{1}{m}\right)\right) \tag{112}$$

and solve for ϵ. We get

$$\delta(d, p_e) = 4\left(\frac{1}{n} + \frac{1}{\mathcal{T}} + \frac{1}{m}\right)^{-1} Q_d^{-1}\left(\frac{p_e}{4}\right). \tag{113}$$

Next, we present the code, which is adapted from the normalized two–part codes (cf. [16] and the references therein). A two–part code is comprised of two–parts. Part one of the code describes a net of parameter vectors in the space of parameter vectors. Part two of the code is the Shannon code for the sequence using the optimal parameter vector from the net. Barron and Cover [14] demonstrated that a net covering of the space exists using d-dimensional ellipsoids. Let the space be partitioned into ellipsoids of the form $\mathcal{S}_{n,\mathcal{T},m}(p_e)$. Then, each sequence is encoded within its respective ellipsoid without regard to the rest of the parameter space. The decoder chooses the decoding ellipsoid using the ML estimate $\hat{\theta}_Y$ and the permissible decoding error probability p_e. As the unknown parameter vector is assumed to follow the least favorable Jeffreys' prior, Lemma 2.3.3 bounds the probability measure of the parameter vectors that are covered by each ellipsoid. As can be seen, the probability measure covered by each ellipsoid is $\mathbf{P}_\theta[\mathcal{S}_{n,\mathcal{T},m}(p_e)]$, which is independent of $\hat{\theta}_Y$. This provides with $-\log \mathbf{P}_\theta[\mathcal{S}_{n,\mathcal{T},m}(p_e)]$ reduction in the redundancy. Also, please note that it is straightforward to show

$$Q_d^{-1}\left(\frac{p_e}{4}\right) \simeq \frac{d}{2}\log e + \log\frac{4}{p_e} \tag{114}$$

Hence,

$$\begin{aligned}\bar{R}_{ED}^{p_e} &\le \frac{d}{2}\log\left(\frac{n}{2\pi e}\right) + \log\int_{\theta\in\Lambda} |\mathcal{I}(\theta)|^{\frac{1}{2}} d\theta \\ &\quad + \log \mathbf{P}_\theta[\mathcal{S}_{n,\mathcal{T},m}(p_e)] + O\left(\frac{1}{n}\right) && (115)\\ &\simeq \frac{d}{2}\log\left(1+\frac{n}{\mathcal{T}}+\frac{n}{m}\right) + \mathcal{F}(d,p_e) + o(1), && (116)\end{aligned}$$

where $\mathcal{F}(d,p_e)$ is defined in (105). This leads to the desired result in Theorem 4.5.4.

$$\begin{aligned}\bar{R}_{ED}^{p_e} &\le \frac{d}{2}\log\left(\frac{n}{2\pi e}\right) + \log\int_{\theta\in\Lambda} |\mathcal{I}(\theta)|^{\frac{1}{2}} d\theta + \log N_\epsilon + O\left(\frac{1}{n}\right). && (117)\\ &\simeq \frac{d}{2}\log\left(1+\frac{2n}{\mathcal{T}}+\frac{2n}{m}\right) + 2 + \log\frac{1}{p_e} && (118)\end{aligned}$$

□

In the case of almost lossless UcompED coding, the following theorem quantifies the performance bounds.

Theorem 4.5.9. *For the minimum average redundancy for lossless UcompED coding,* **(a)** *in the case of $\mathcal{T} = o(n)$, $\bar{R}_{ED}^{p_e}$ is given by*

$$\bar{R}_{ED}^{p_e} = \bar{R}^0 + O(1),$$

and **(b)***, in the case of $\mathcal{T} = \omega(n)$, $\bar{R}_{ED}^{p_e}$ is bounded by*

$$(1-p_e)\bar{R}_{ED}^0 - h(p_e) - p_e H_n(\theta^{(2)}) \le \ \bar{R}_{ED}^{p_e} \ \le \ \bar{R}_{ED}^0.$$

Proof. The proof of Part (a) is straightforward by considering Lemma 4.3.5 and the proof of Part (b) follows the lines of the proof of Theorem 4.5.1. □

4.6 Discussion

In this section, we provide some discussion on the significance of the results for different coding strategies for the case $\mathcal{T} = \omega(n)$. We discuss the strictly lossless case followed by two examples that illustrate the impact of the source parameter correlation on the minimum average redundancy of the almost lossless compression schemes.

4.6.1 Strictly Lossless Coding

In the case of Ucomp, (94) determines the achievable minimum average redundancy for the compression of a sequence of length n encoded without regard to the side information sequence y^m. Hence, Ucomp is regarded as the benchmark for the performance of UcompD, UcompE, and UcompED, which use the side information sequence.

For strictly lossless UcompE and UcompD, according to Propositions 4.4.1 and 4.4.2, the side information provided by y^m from S_1 does not provide any performance improvement in the universal compression of x^n assuming the side information is not present in either the encoder or the decoder. In other words, the best that S_2 can do for the strictly lossless compression of x^n is to simply apply a strictly lossless universal compression without side information (i.e., Ucomp).

When the side information is present at both the encoder and the decoder, from Lemma 4.3.3 it is expected that the performance of strictly lossless UcompED coding on the universal compression of x^n would be improved with respect to Ucomp, which is quantified by Theorem 4.4.3 as given by $\hat{R}(n, m, \mathcal{T})$ in (95). We will discuss the dependence of $\hat{R}(n, m, \mathcal{T})$ on the sequence length n, size of the side information m, and the degree of correlation $\mathcal{T}$ in Section 4.6.2.

In summary, for the strictly lossless case, only UcompED offers improvement over Ucomp but it is not practical for the universal compression of distributed sources as it requires the encoders to communicate.

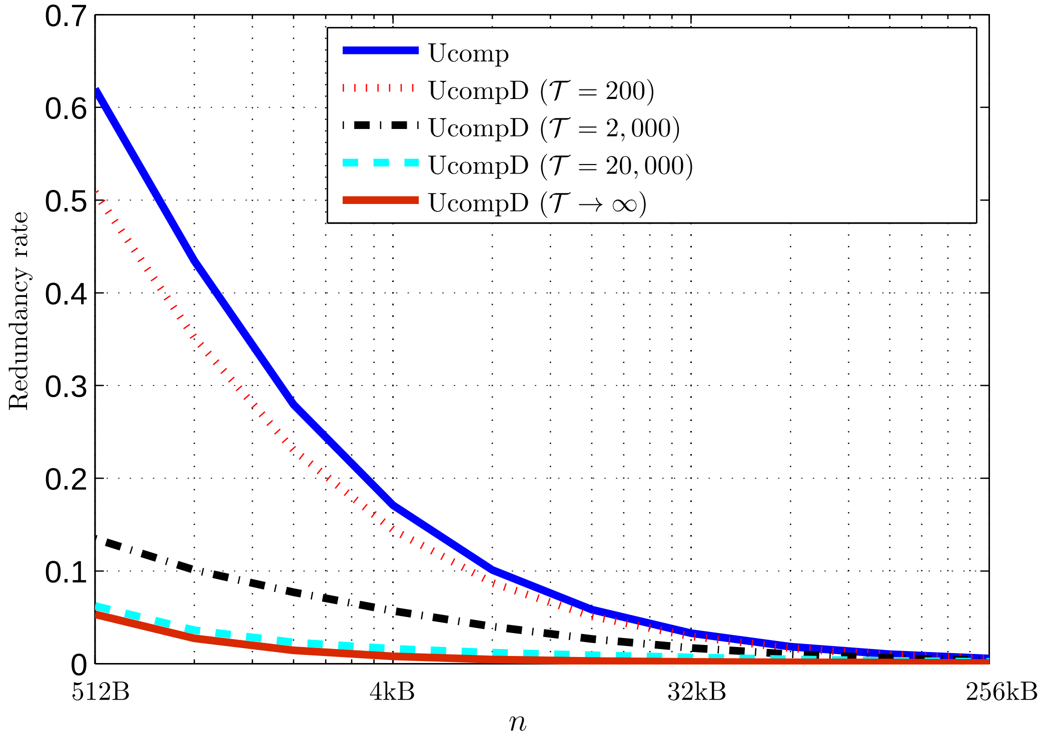

Figure 13: The redundancy rate of the three coding strategies of interest in the DSC-CPV problem for different correlation parameter vectors, where the memory size is $m = 32$kB. The two memoryless sources have alphabet size $|\mathcal{A}| = 256$.

4.6.2 The Degree of Correlation and the Memory Size

Next, we investigate the impact of the degree of correlation $\mathcal{T}$ and the size of the side information sequence m on the main redundancy term $\hat{R}(n, m, \mathcal{T})$ in (95), which constitutes the main term in most of the results. In fact, $\hat{R}(n, m, \mathcal{T})$ can be rewritten as

$$\hat{R}(n, m, \mathcal{T}) = \frac{d}{2} \log \left(1 + \frac{n}{m^\star}\right), \tag{119}$$

where $m^\star$ is defined as

$$\frac{1}{m^\star} \triangleq \frac{1}{m} + \frac{1}{\mathcal{T}}. \tag{120}$$

Therefore, if n, m, and $\mathcal{T}$ are thought of as the values of resistances, $m^\star$ would be the *parallel* resistance of the elements m and $\mathcal{T}$. Thus, $\hat{R}(n, m, \mathcal{T})$ is determined

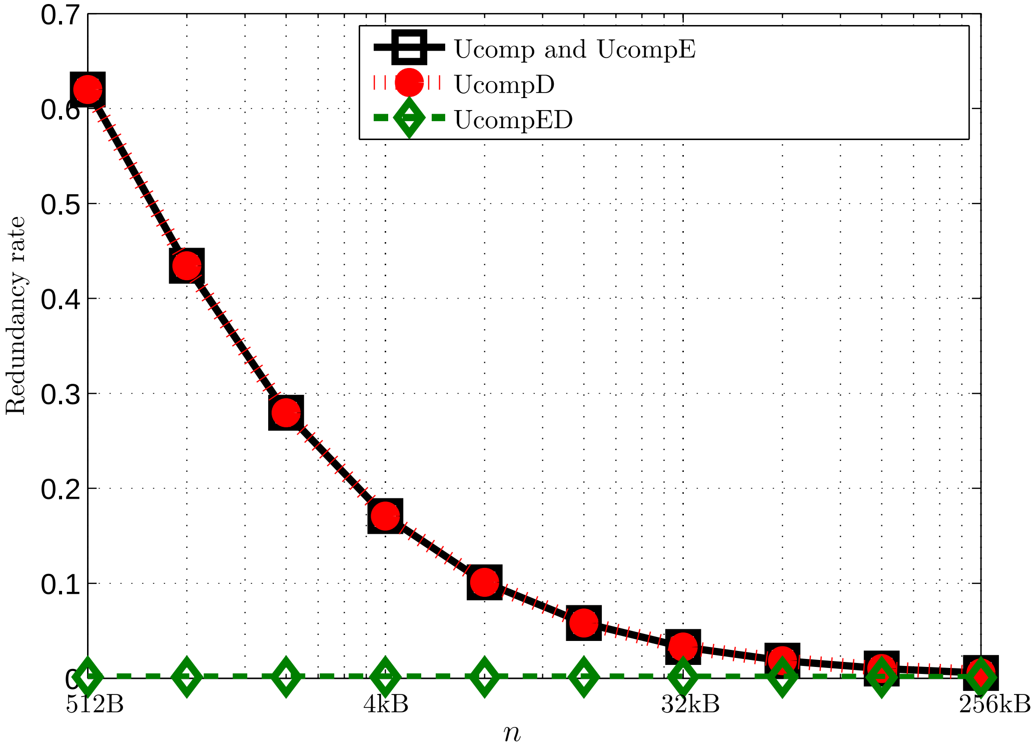

Figure 14: The redundancy rates of the four different codings of interest for the strictly lossless case (i.e., $p_e = 0$), where the memory size is $m = 32$kB. The two identical memoryless sources have alphabet size $|\mathcal{A}| = 256$.

by the ratio $\frac{n}{m^\star}$. When m and $\mathcal{T}$ are both finite, increasing one of them beyond a certain limit does not reduce the redundancy as the parallel resistance converges to the smaller of the two.

It is straightforward to see that $\hat{R}(n, m, \mathcal{T})$ decreases as the memory size m grows and for very large memory (i.e., $m \to \infty$) converges to

$$\hat{R}(n, \infty, \mathcal{T}) = \frac{d}{2} \log \left(1 + \frac{n}{\mathcal{T}}\right), \tag{121}$$

which is merely a function of the degree of correlation between the two sources. Thus, if a sufficiently large memory m is available the performance limitation will become a function of $\frac{n}{\mathcal{T}}$. Furthermore, when $\mathcal{T} \to \infty$, i.e., identical sources as studied in [18], then $\hat{R}(n, m, \infty)$ vanishes as $m \to \infty$.

Figure 13 demonstrates the redundancy rate for different values of $\mathcal{T}$ for the case

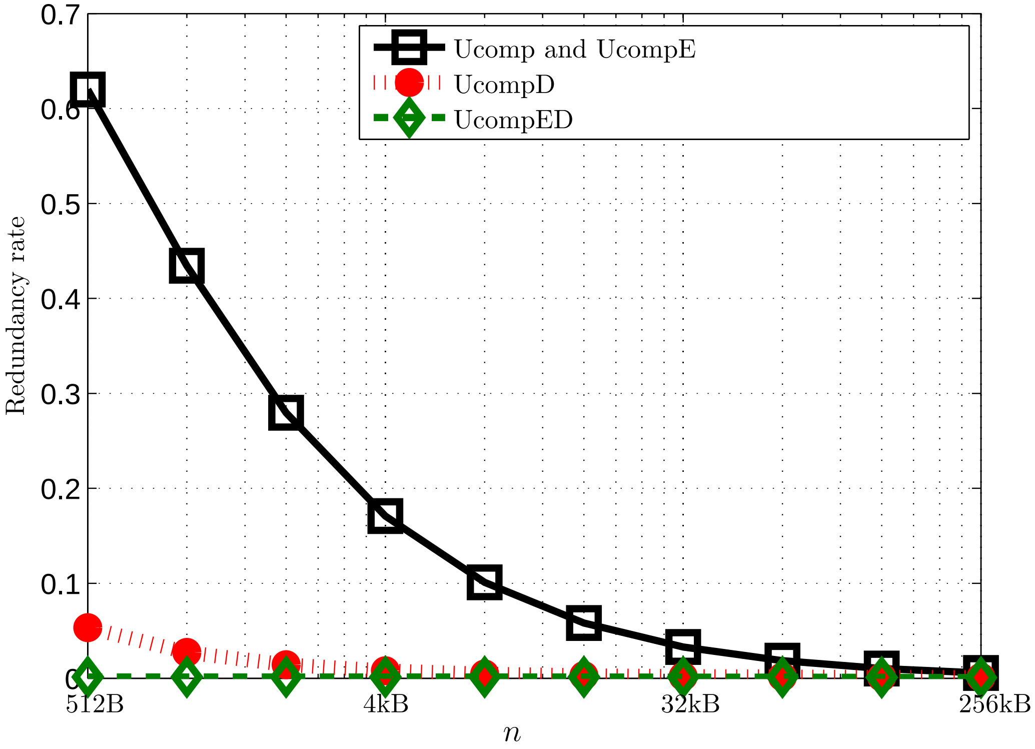

Figure 15: The redundancy rates of the four different codings of interest for the almost lossless case with permissible error probability $p_e = 10^{-8}$, where the memory size is $m = 32$kB. The two identical memoryless sources have alphabet size $|\mathcal{A}| = 256$.

of memoryless sources with alphabet size $|\mathcal{A}| = 256$ when m is sufficiently large and hence $m^\star$ is dominated by $\mathcal{T}$. As can be seen, as the correlation between the two parameter vectors increases, the redundancy decreases and eventually converges to that of the identical source parameter vectors.

4.6.3 Identical Source Parameter Vectors (Case $\mathcal{T} \to \infty$)

In this special case, we assume that $\mathcal{T} \to \infty$ and hence $\theta^{(2)} = \theta^{(1)}$. Then, the performance of strictly lossless UcompED coding and the almost lossless UcompD coding is quantified by $\hat{R}(n, m, \infty)$ given by

$$\hat{R}(n, m, \infty) = \frac{d}{2} \log \left(1 + \frac{n}{m}\right), \tag{122}$$

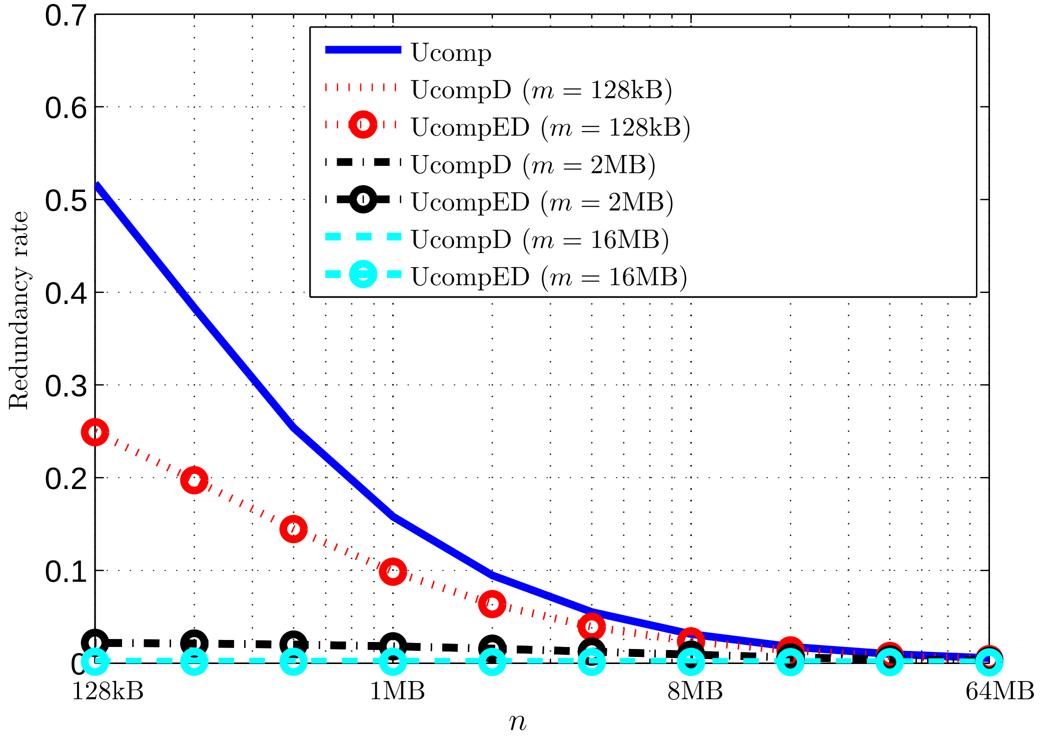

Figure 16: The redundancy rate of the three coding strategies of interest for different memory sizes, where the permissible error probability is $p_e = 10^{-8}$. The two identical sources are first-order Markov with alphabet size $|\mathcal{A}| = 256$.

which gives back the main result in [17], where the identical source parameters were studied. If we further consider the redundancy for large m, we observe that $\underline{R}^0_{\text{UcompED}}(n, \infty, \infty) = 0$. In other words, since the identical parameter vectors will be known to both the encoder and the decoder, the average codeword redundancy vanishes and the optimal code for known source parameter vectors is obtained. In this case, the fundamental limits are those of known source parameter vectors and universality no longer imposes a compression overhead.-

Figure 14 demonstrates the redundancy rate for the three coding strategies for memoryless sources with identical source parameter vectors (i.e., $\theta^{(2)} = \theta^{(1)}$) and alphabet size $|\mathcal{A}| = 256$. As can be seen, for both Ucomp and UcompED, the lower bounds on the lossless coding closely follow the strictly lossless coding. Further, using

Fact 4.3.2, we conclude that little improvement is obtained over strictly lossless case when using reasonably small permissible error probability. On the other hand, the lossless UcompD coding has the potential to significantly improve the performance over strictly lossless case (as can be seen from the upper bound plotted in Figure 14). For example, when $n = 512$B and $m = 32$kB, if the permissible error probability is $p_e = 10^{-8}$, the redundancy rate of UcompD coding is upper bounded by approximately 0.05, whereas that of the strictly lossless case is almost 0.62. Note that, however, the development of a practical coding scheme that achieves the bound remains an open problem.

Figure 16 demonstrates the redundancy rate for two identical first-order Markov sources with alphabet size $|\mathcal{A}| = 256$ for different memory sizes. As can be seen the redundancy rate decreases as m grows. For sufficiently large m, the redundancy rate approaches zero as previously discussed. Further, comparing Figure 16 and Figure 14, when the number of source parameter vectors is relatively large (i.e., Markov vs. memoryless), even with small permissible error probability, UcompD performs fairly close to UcompED. Finally, UcompD by far outperforms Ucomp in the compression of short to medium length sequences with reasonable permissible error probability, justifying the usefulness of UcompD.

4.7 Conclusion

In this chapter, the problem of universal compression of two distributed parametric sources with correlated parameter vectors was introduced and studied. A correlation model for the two source parameter vectors was formally defined, which departs from the nature of the correlation in the SW framework. Involving two correlated sources, the minimum average redundancy for four different coding strategies (based on whether or not the side information was available to the encoder and/or the decoder) was investigated. These strategies are (1) universal compression without side information, (2) universal compression with encoder side information, (3) universal compression with decoder side information, and (4) universal compression with encoder-decoder side information.

It was proved that as the length of the side information sequence grows to infinity and the source models are almost equal, coding strategy 3 achieves a vanishing redundancy. We further demonstrated that there is a gap between the performance of the strategies 2 and 3. It was further demonstrated that for short to medium length sequences with plausible permissible error probability, strategy 2 by far outperforms strategy 1, and hence, justifying the usefulness of strategy 2 by providing a constructive coding for strategy 2. In summary, this chapter demonstrated that the side information at the encoder and/or the decoder in the network can help to noticeably improve the performance of the universal compression on distributed parametric sources with correlated parameter vectors.

CHAPTER V

CLUSTER-BASED MEMORY-ASSISTED COMPRESSION

Universal compression aims at reducing the average number of bits required to describe a sequence from an unknown source from a family of sources, while good performance is desired for most of the sources in the family. However, it often needs to observe a very long sequence so that it can effectively learn the existing patterns in the sequence for efficient compression. Therefore, universal compression performs poorly on relatively small sequences [16, 52] where sufficient data is not available for learning of the statistics and training of the compressor. On the other hand, the presence of side information has proven to be useful in several source coding applications (cf. [79, 89, 54, 20, 55] and the references therein). In particular, the impact of side information on *universal* compression has also been shown to be useful (cf. [34, 47, 18, 17]). However, to the best of the authors' knowledge, the impact of side on information on the universal compression of a mixture of parametric sources has not been explored in the literature.

In Chpater 3, we proposed universal compression of network packets using network memory, where the common memory between the encoder router and the decoder router was used as the side information to improve the performance of universal compression on network packets. As each packet may be generated by a different source, a realistic modeling of the network traffic requires to consider the content server to be a mixture of parametric sources [70]. This motivates us to study the universal compression of sequences from a mixture source using common side information between the encoder and the decoder.

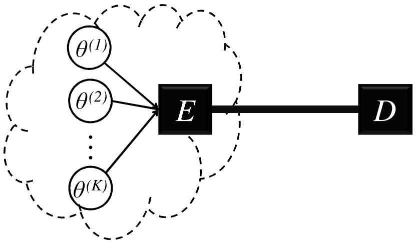

Figure 17: The basic scenario of universal compression with side information for a mixture source.

Although the problem formulation is inspired from the network traffic compression, universal compression of a mixture source with side information finds applications in a wide variety of problems, such as data storage systems, and migration of virtual machines, where the compression of data before transmission results in improved performance. As shown in Figure 17, we assume that each sequence (e.g., network packet) is a sample of length n from a mixture of K parametric sources. We consider each source $i \in [K] \triangleq \{1, \ldots, K\}$ in the mixture as a parametric source with a d_i-dimensional parameter vector $\theta^{(i)}$ such that $\theta^{(i)}$ is drawn independently from Jeffreys' prior. As was discussed in Section 2.1, Jeffreys' prior is assumed to ensure the worst-case performance analysis. We further assume that each output sequence from this mixture source is chosen from $\theta^{(S)}$, where the index S of the source is chosen at random from $[K]$ according to the probability law $\mathbf{w} = (w_1, \ldots, w_K)$. We consider the scenario where T sequences from the mixture source are shared as side information between the encoder E and the decoder D in Figure 17. The first objective is to derive the average redundancy incurred in the *optimal* universal compression with side information where optimality is in the sense of minimizing the average redundancy as a function of n, K, and T. Note that in the network compression application, E and D, in Figure 17, can be thought of as two routers inside the network that have a shared common memory of sequences from the mixture source.

In the previous chapters, we derived the optimal universal compression performance with side information for a single source, i.e., $K = 1$; we proved that significant improvement is obtained from the side information in the universal compression of small sequences when sufficiently large side information is available. In [18], we extended the setup to finite K which is known to the encoder and the decoder a priori. We further assumed that the indices of the sources that generated the side information sequences are also known to both the encoder and the decoder (i.e., perfect clustering of the memory based on the mixture index is possible). We demonstrated that the universal compression using clustering of the side information by the source indices offers significant improvement over universal compression without side information. Inspired from this, in [70], we developed a clustering algorithm for the universal compression with side information based on the Hellinger distance of the sequences and showed its effectiveness on real network traffic traces. However, it remained an open problem to determine the *optimal* strategy that utilizes the side information in the sense that, given the side information, the minimum codeword length in the universal compression of a new sequence from the mixture source using side information is attained.

In this chapter, we generalize the setting of Chapter 3. We let K grow with n; we drop the assumption that the indices of the sources that generated the side information sequences are known; we relax the assumption that K is known to both the encoder and the decoder a priori. We provide theoretical justification of the clustering solution supplemented by computer simulations. We further develop and study a clustering algorithm tailored to compression. Our contributions in this work can be summarized as the following.

- The smallest achievable average redundancy incurred in universal compression of a random sequence of length n from a mixture source given that the encoder and the decoder have access to a shared side information of T sequences (each of length n from the mixture of K parametric sources) is characterized and verified using computer simulations.

- It is demonstrated that the performance of the optimal universal compression with side information almost surely coincides with that of the universal compression with perfect clustering of the memory, and hence, it is concluded that clustering is optimal for universal compression with side information.

- A parametric clustering strategy based on the K-means algorithm is provided for the memory-assisted compression that aims at grouping the side information sequences that share similar statistical properties. A newly generated packet by the mixture source is classified into one of the clusters for compression. It is demonstrated through experiments performed on real network traffic traces that the proposed algorithm is effective.

The rest of this chapter is organized as follows. In Section 5.1, we present the formal definition of the problem. In Section 5.2, we derive the entropy of the mixture source, which serves as a lower limit on the average codeword length. In Section 5.3, we provide the main results on the universal compression of mixture sources with and without side information and discuss their implications. In Section 5.4, we present the parametric clustering algorithm used for the compression of the mixture sources. In Section 5.5, we provide simulation results that support our theoretical results on the compression of the mixture sources on man-made data as well as data gathered from real network traffic traces. In Section 5.6, we provide the technical analysis of the results. Finally, Section 5.7 concludes this chapter.

5.1 Problem Setup

In this section, we present the setup of the universal compression with common side information at the encoder and the decoder. Let a parametric source be defined using a d-dimensional parameter vector $\theta = (\theta_1, ..., \theta_d) \in \Lambda_d$ that is a priori unknown, where d denotes the number of the source parameters and $\Lambda_d \subset \mathbb{R}^d$ is the space of d-dimensional parameter vectors of interest. Denote $R_{n,d}(l_n, \theta)$ as the expected redundancy of the code c_n with length function l_n on a sequence of length n for the parameter vector θ.

Let $\Delta \triangleq \left\{\theta^{(i)}\right\}_{i=1}^{K}$ denote the set of $K \triangleq |\Delta|$ parameter vectors of interest where $\theta^{(i)} \in \Lambda_{d_i}$ is a d_i-dimensional parameter vector. Note that we let K deterministically scale with n. Let $d_{\max} \triangleq \max\{d_1, \ldots, d_K\}$ denote the maximum dimension of the parameter vectors, where we assume that $d_{\max} = O(1)$, i.e., $d_{\max}$ is finite. We further assume that for any $d < d'$, we have $\Lambda_d \subset \Lambda_{d'}$, and hence, Δ consists of K points on the space $\Lambda_{d_{\max}}$.

We assume that $\forall i \in [K]$, we have $\theta^{(i)} = (\theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_{d_i}^{(i)})$ is chosen at random according to the Jeffreys' prior on the d_i-dimensional parameter space Λ_{d_i}. In this setup, as in Figure 17, the source is a mixture of K parametric sources $\mu_{\theta^{(1)}}, \ldots, \mu_{\theta^{(K)}}$, where for all $i \in [K]$, $\theta^{(i)}$ is a d_i-dimensional unknown parameter vector. For the generation of each sequence of length n, the generator source is selected according to the probability law $\mathbf{w} = (w_1, \ldots, w_K)$ from the mixture, i.e., Δ. In other words, $p(\theta|\Delta) = \sum_{i=1}^{K} w_i \delta(\theta - \theta^{(i)})$, where $\theta^{(i)}$ follows Jeffreys' prior on Λ_{d_i} and w_i is the probability that the sequence is generated by source $\theta^{(i)}$ in the mixture. Please note that the random set Δ (which is unknown a priori) is randomly generated once according to Jeffreys' prior and is used thereafter for the generation of all sequences from the mixture source. Let S be a random variable that determines the source index, and hence follows the distribution $\mathbf{w}$ over $[K]$, i.e., $\mathbf{P}[S = i] = w_i$. Then, by definition, we have $\theta = \theta^{(S)}$ given Δ. Unlike Δ that is generated once, S is chosen with $\mathbf{w}$ every time a new sequence is generated. Let the mixture entropy $H(\mathbf{w})$ be defined as $H(\mathbf{w}) = -\sum_{i \in [K]} w_i \log w_i$.[1]

[1] We define entropy $H(\mathbf{r})$ for any vector $\mathbf{r}$ such that $\sum_i r_i = 1$ in the same manner throughout

We consider the following scenario. We assume that, in Figure 17, both the encoder E and the decoder D have access to a common side information of T previous sequences (indexed by $[T]$) from the mixture of K parametric sources, where each of these sequences is independently generated according to the above procedure. Let $m \triangleq nT$ denote the aggregate length of the previous T sequences from the mixture source.[2] Further, denote $\mathbf{y}^{n,T} = \{y^n(t)\}_{t=1}^{T}$ as the set of the previous T sequences shared between E and D, where $y^n(t)$ is a sequence of length n generated from the source $\theta^{S(t)}$ and $S(t)$ follows $\mathbf{w}$ on $[K]$. In other words, $y^n(t) \sim \mu_{\theta(S(t))}$. Further, denote $\mathbf{S}$ as the vector $\mathbf{S} = (S(1), ..., S(T))$, which contains the indices of the sources that generated the T previous side information sequences.

Let $\hat{l}(x^n, \mathbf{y}^{n,T})$ denote a generic length function that utilizes the side information $\mathbf{y}^{n,T}$ in the compression of a new sequence x^n. The objective is to analyze the average redundancy in the compression of a new sequence x^n that is independently generated by the same mixture source with source index Z (which also follows $\mathbf{w}$). We investigate the fundamental limits of the universal compression with side information ($\mathbf{y}^{n,T}$) that is shared between the encoder and the decoder and compare with that of the universal compression without side information of the previous sequences. It is straightforward to verify that $H(X^n|\mathbf{Y}^{n,T})$ and $H(X^n)$ for different values of the sequence length n, memory (side information) size $m = nT$, the weight of the mixture $\mathbf{w}$, and the dimensions of the parameter vectors $\mathbf{d}$ serve as two of the main fundamental limits of the compression, which seek the minimum number of bits required to represent a random sequence X^n when $\mathbf{Y}^{n,T}$ is present (at both the encoder and the decoder) or not.

the chapter.

[2] For simplicity of the discussion, we consider the lengths of all sequences to be equal to n. However, most of the results are readily extendible to the case where the sequences are not necessarily equal in length.

5.2 Entropy of the Mixture Source

Before we state the main results of this work, we need to derive the entropy of the mixture source.

Definition. Let $H_n(\Delta, Z) \triangleq H(X^n|\Delta, Z)$ be defined as the entropy of a random sequence X^n from the mixture source given that the source parameters are known to be the set Δ and the index of the source that has generated the sequence (i.e., Z) is also known.[3] In other words, the parameter vector $\theta^{(Z)}$ associated with sequence X^n is known. Then, in this case, by definition

$$H_n(\Delta, Z) = \sum_{i=1}^{K} w_i H_n(\theta^{(i)}), \tag{123}$$

where $H_n(\theta^{(i)})$ is the entropy of source $\mu_{\theta^{(i)}}$ given $\theta^{(i)}$ defined in (1). Please note that $H_n(\Delta, Z)$ is *not* the achievable performance of the compression. It is merely introduced here so as to make the presentation of the results more convenient.

Let the set Δ be written as the following.

$$\Delta = \cup_{d=1}^{d_{\max}} \Delta_d, \tag{124}$$

where Δ_d is the set of the d-dimensional parameter vectors in Δ. Further, let $K_d \triangleq |\Delta_d|$ be the number of parameter vectors in set Δ_d. In other words, K_d is the number of sources of dimension d in the mixture source. Hence, $\sum_{d=1}^{d_{\max}} K_d = K$. Now, we can relabel the elements in Δ according to their parameter vectors. Let $\Delta_d = \{\theta^{(d,1)}, \ldots, \theta^{(d,K_d)}\}$. Denote $\mathbf{w}_d = (w_{d,1}, \ldots, w_{d,K_d})$ as the weight of the d-dimensional parameter vectors. Further, let $v_d \triangleq \sum_{i=1}^{K_d} w_{d,i}$ be the aggregate weight of all d-dimensional parameter vectors and denote $\mathbf{v} \triangleq (v_1, \ldots, v_{d_{\max}})$. Let $\hat{\mathbf{w}}_d \triangleq \mathbf{w}_d / v_d$, i.e., we have $\hat{w}_{d,i} \triangleq w_{d,i}/v_d$, for $1 \leq i \leq K_d$.

[3] We assume that the random set of parameter vectors is generated once and used for the generation of all sequences of length n thereafter. Therefore, throughout the chapter, whenever we assume that Δ is given, we mean that the set of the parameter vectors is known to be the set Δ.

Hence, $H_n(\Delta, Z)$ can be rewritten as

$$\begin{aligned} H_n(\Delta, Z) &= \sum_{d=1}^{d_{\max}} \sum_{i=1}^{K_d} w_{d,i} H_n(\theta^{(d,i)}) \\ &= \sum_{d=1}^{d_{\max}} v_d \sum_{i=1}^{K_d} \hat{w}_{d,i} H_n(\theta^{(d,i)}). \end{aligned} \tag{125}$$

Next, we derive the entropy of the mixture source (which sets the asymptotic fundamental lower limit on the codeword length for the known source parameters case), i.e., when Δ is known. Define $H_n(\Delta) \triangleq H(X^n|\Delta)$.

Theorem 5.2.1. *The entropy of the mixture source is given by*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{v}) + \sum_{d=1}^{d_{max}} v_d H_d + O\left(\frac{1}{n}\right) \ \ a.s.,^{4,5}$$

where H_d is given by

$$H_d = \begin{cases} H(\hat{\mathbf{w}}_d) & \text{if } H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n \\ \bar{R}_{n,d} & \text{if } H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n \end{cases},^{6} \tag{126}$$

and $\bar{R}_{n,d}$ is given by (9)*.*

Proof. See Section 5.6 for the proof. □

Remark. Theorem 5.2.1 determines the entropy of the mixture source, which corresponds to the minimum codeword length when the parameter vectors in the set Δ are known to the encoder and the decoder (i.e., non-universal compression). Please note that $H_n(\Delta)$ also serves as a trivial lower bound on the codeword length for the case of universal compression (i.e., unknown parameter vectors). Please note that for sufficiently low-entropy $\hat{\mathbf{w}}_d$ or for sufficiently small K_d, the price of describing the

[4] *An event A happens a.s. (almost surely) if and only if* $\mathbb{P}[A] = 1$.

[5] *Please note that the sample space is the set of all source parameter vectors* $\Delta = \left\{\theta^{(i)}\right\}_{i=1}^{K}$ *such that $\theta^{(i)}$ is drawn independently from Jeffreys' prior.*

[6] $f(n) \prec g(n)$ *if and only if* $\lim_{n\to\infty} \frac{f(n)}{g(n)} < 1$.

d-dimensional parameter vectors is, on average, equal to $H(\hat{\mathbf{w}})$, which corresponds to describing the respective source parameter vector in the encoder.

The following corollary describes the entropy when the number of source parameter vectors are sufficiently small.

Corollary 5.2.2. *If $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, then*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{w}) + O\left(\frac{1}{n}\right) \quad a.s. \tag{127}$$

Proof. Since $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, we have $K_d = O\left(n^{\frac{d}{2}-\epsilon}\right)$ for some $\epsilon > 0$, and hence, we have $H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n$. Thus, $H_d = H(\hat{\mathbf{w}}_d)$ for all $1 \le d \le d_{\max}$. The proof is completed by noting that

$$H(\mathbf{w}) = H(\mathbf{v}) + \sum_{d=1}^{d_{\max}} v_d H(\hat{\mathbf{w}}_d).$$

□

Remark. According to the corollary, when $K = O\left(n^{\frac{1}{2}-\epsilon}\right)$ for some $\epsilon > 0$, the optimal coding strategy (when the source parameters are known) almost surely would be to encode the source index Z and then use the optimal code (e.g., Huffman code) associated with parameter $\theta^{(Z)}$ for sequences of length n to encode the sequence x^n. In fact, if $H(\mathbf{w}) \prec \frac{d}{2}\log n$, then the cost of encoding the parameter is asymptotically smaller than the cost of universally encoding the parameter and hence it is beneficial to encode the parameter vector using an average of $H(\mathbf{w})$ bits. Further, if $K = 1$, then $\Delta = \theta^{(1)}$ and $Z = 1$ would be deterministic. Hence, $H_n(\Delta) = H_n(\Delta, Z) = H_n(\theta^{(1)})$, which was introduced in (1) as the average compression limit for the case of a single known source parameter vector.

Corollary 5.2.3. *If $H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n$ for all $1 \le d \le d_{max}$ such that $v_d > 0$, then*

$$H_n(\Delta) = H_n(\Delta, Z) + H(\mathbf{v}) + \sum_{d=1}^{d_{max}} v_d \bar{R}_{n,d} + O\left(\frac{1}{n}\right) \quad a.s. \tag{128}$$

Proof. The proof is very similar to the previous corollary and is omitted for brevity.

□

Remark. According to the corollary, in the case where the number of sources in the mixture is very large, the mixture entropy converges to $H_n(\Delta, Z)$ plus $H(\mathbf{v})$ plus the weighted average of the $\bar{R}_{n,d}$ terms (which are exactly the average maximin redundancy in the *universal* compression of parametric sources with d *unknown* parameters given in Theorem 2.1.4). At the first glance, it may seem odd that the codeword length in the case of *known* source parameter vectors incurs a term that is associated with the universal compression of a source with an *unknown* parameter vector. A closer look, however, reveals that in this case the cost of encoding the source index of a d-dimensional parameter vector surpasses the cost of universally encoding the source parameter vector. Hence, intuitively, it no longer makes sense to encode the d-dimensional parameter vector for the compression of the sequence x^n using an average of $H(\hat{\mathbf{w}}_d)$ bits. More rigorously speaking, as is shown in the proof of Theorem 5.2.1 in Section 5.6, the probability distribution of x^n given $\theta \in \Delta_d$ would converge to the probability distribution of x^n when the source has one *unknown* d-dimensional parameter vector that follows Jeffreys' prior. This in turn results in the $\bar{R}_{n,d}$ term in the compression performance.

5.3 Fundamental Limits of Universal Compression for Mixture Sources

In this section, we state our main results on the fundamental limits of universal compression for mixture sources with and without side information. The proofs are deferred to Section 5.6. In order to see the impact of the universality and side information on the compression performance, i.e., to investigate the impact of Δ being unknown, we will need to analyze and compare the average codeword length of the following important schemes described in the sequel.

- Ucomp: Universal compression, which is the conventional compressed based solution.

- UcompSM: Simple universal compression with side information (common memory between the encoder and the decoder), which treats the side information as if it were generated from a single parametric source.

- UcompPCM: Universal compression with perfectly clustered side information (based on the source indices), which assumes that the source indices of the side information sequences can be determined using an oracle, and hence, only the relative side information is used toward the compression of a new sequence.[7]

- UcompOM: Optimal universal compression with side information, which optimally utilizes the side information sequence $\mathbf{y}^{n,T}$ to minimize the average redundancy.

- UcompCM: Universal compression with clustering of the side information, which is the practical clustering-based scheme proposed in this work and shall be described in Section 5.4.

Definition. We refer to Ucomp as the universal compression without side information, in which a sole universal compression is applied on the sequence x^n without regard to the side information sequence $\mathbf{y}^{n,T}$. We further refer to $R(n, \mathbf{w}, \mathbf{d})$ as the average redundancy of the universal compression of a sequence of length n (in our problem setup described in Section 5.1). In other words,

$$R(n, \mathbf{w}, \mathbf{d}) \triangleq H(X^n) - H_n(\Delta). \tag{129}$$

[7] Please note that UcompPCM scheme is not practically interesting as the oracle that provides the index of the sequence is usually not available.

Please note that $R(n, \mathbf{w}, \mathbf{d})$ also implicitly depends on the prior used on the parameter vectors, which is Jeffreys' prior in this work. It is straightforward to show that Jeffreys' prior is also the capacity achieving prior for the mixture, i.e., it maximizes $R(n, \mathbf{w}, \mathbf{d})$.

Theorem 5.3.1. *In the case of* Ucomp*, we have*

$$R(n, \mathbf{w}, \mathbf{d}) = \sum_{i=1}^{d_{max}} v_d(\bar{R}_{n,d} - H_d) + O\left(\frac{1}{n}\right) \text{ a.s.},$$

where H_d is defined in (126)*.*

Proof. See Section 5.6 for the proof. □

Remark. According to Theorem 5.3.1, in the universal compression of a sequence of length n from the mixture source, the main term of the redundancy scales as the weighted average of $(\bar{R}_{n,d} - H_d)$ terms. This can be significantly large if $H(\mathbf{w}_d)$ is much smaller than $\frac{d}{2}\log n$. Again, if $K = 1$, we have $R(n, 1, d) = \bar{R}_{n,d}$; this is exactly the average maximin redundancy in the case of one unknown d-dimensional source parameter vector described in Theorem 2.1.4.

Theorem 5.3.1 also suggests that independently from K and $H(\mathbf{w})$, the price to be paid for universality is given by $\bar{R}_{n,d}$ on top of $H_n(\Delta, Z)$, i.e., the entropy when Δ and Z are known. In other words, $H(X^n) - H_n(\Delta, Z)$ scales like $\bar{R}_{n,d}$ (the price of universal compression of a sequence of length n from a single source with an unknown d-dimensional parameter vector that follows Jeffreys' prior).

Corollary 5.3.2. *If $H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n$ for all $1 \leq d \leq d_{max}$ such that $v_d > 0$, then*

$$R(n, \mathbf{w}, \mathbf{d}) = O\left(\frac{1}{n}\right) \text{ a.s.}$$

Proof. Since $H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n$ for all $1 \leq d \leq d_{\max}$, we have $H_d = \bar{R}_{n,d}$ for all $1 \leq d \leq d_{\max}$ such that $v_d > 0$. Hence, the main redundancy term in Theorem 5.3.1 vanishes and the proof is completed. □

Remark. According to the corollary, for large K, we almost surely expect no extra redundancy associated with universality on top of the mixture entropy. This is not surprising as even in the case of *known* source parameter vectors, as given by Theorem 5.2.1, the redundancy converges to the weighted average of the redundancies for a d-dimensional *unknown* source parameter vector that follow Jeffreys' prior. Therefore, there is no extra penalty when the source parameter vectors are indeed unknown.

Definition. We refer to UcompSM as the simple universal compression with side information, in which the encoder E and the decoder D (in Figure 17) both have access to the memorized sequence $\mathbf{y}^{n,T}$ from the mixture source; the sequence $\mathbf{y}^{n,T}$ is used to estimate using the minimax estimator (which is the KT-estimator [47] in the case of memoryless sources) the source parameter vector as if $\mathbf{y}^{n,T}$ were generated by a single parametric source with an unknown parameter vector. The estimated source parameter is then used for the compression of the sequence x^n. We further refer to $R_{\text{SM}}(n, m, \mathbf{w}, \mathbf{d})$ as the average redundancy of UcompSM.

Theorem 5.3.3. *In the case of* UcompSM*, we have*

(a) *if* $K = 1$*, then*

$$R_{\text{SM}}(n, m, 1, d) = \frac{d}{2} \log \left(1 + \frac{n}{m}\right) + O\left(\frac{1}{n}\right).$$

(b) *If* $K \geq 2$*, then*

$$R_{\text{SM}}(n, m, \mathbf{w}, \mathbf{d}) = \Theta(n) \text{ a.s.}^{8}$$

Proof. See Section 5.6 for the proof. □

Remark. According to Theorem 5.3.3, when $K \geq 2$, then we have $R_{\text{SM}}(n, m, \mathbf{w}, \mathbf{d}) = \Theta(n)$ with probability one, i.e., when a source parameter vector is estimated for the

[8] *$f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $g(n) = O(f(n))$.*

mixture source, almost surely, the redundancy of UcompSM is asymptotically worse than that of Ucomp (universal compression without side information) since the latter scales as $\frac{d}{2}\log n$. Therefore, it makes no sense to use UcompSM when the data are generated from a mixture source ($K \geq 2$). We shall also see some discussion on validation of this claim based on simulations in Sec. 5.5. Note that the case $K = 1$ corresponds to the simple parametric source with side information that is treated in detail in Chapter 3. Obviously, good performance is obtained in this case as the scheme is optimal for $K = 1$ in the sense of achieving the average minimax redundancy [34].

Definition. We refer to UcompPCM as the universal compression with perfectly clustered side information sequence $\mathbf{y}^{n,T}$, which is shared between the encoder E and the decoder D. Further, it is assumed that E and D have access to an oracle that can determine the index of the source that generated each sequence. Hence, the index sequence $\mathbf{S}$ of the memorized sequences is known to both E and D and the index Z of the sequence x^n to be compressed is known to E. Then, E and D cluster the side information sequences according to $\mathbf{S}$ and use the minimax estimator to estimate the source parameter vector associated with each cluster; the encoder E classifies the sequence x^n to the respective cluster using the oracle and encodes the sequence only using the side information provided by the estimated parameter vector of the respective cluster.

Theorem 5.3.4. *In the case of* UcompPCM*, we have*

$$R_{\mathrm{PCM}}(n, m, \mathbf{w}, \mathbf{d}) = \sum_{d=1}^{d_{max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i} \hat{R}_{d,i} + O\left(\frac{1}{n}\right) \text{ a.s.},$$

where

$$\hat{R}_{d,i} = \begin{cases} \frac{d}{2}\log\left(1 + \frac{n}{\hat{w}_{d,i}m}\right) & \text{if } H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n \\ 0 & \text{if } H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n \end{cases}. \tag{130}$$

Proof. See Section 5.6 for the proof. □

Remark. Theorem 5.3.4 characterizes the redundancy of the universal compression with perfectly clustered side information. It is straightforward to observe that for sufficiently large m, the redundancy of UcompPCM becomes very small. However, please note that UcompPCM is impractical in most situations as the oracle that provides the source index is not available. Furthermore, UcompPCM is also not necessarily optimal for all n. As an important special case if $K = 1$, then $R_{\text{PCM}}(n, m, \mathbf{w}, \mathbf{d}) = \frac{d}{2} \log\left(1 + \frac{n}{m}\right) + O\left(\frac{1}{n}\right)$, which reduces to Theorem 2 of [17] regarding the average minimax redundancy for the case of a single source with an unknown parameter vector.

Corollary 5.3.5. *Regardless of* $\mathbf{w}$ *and* $\mathbf{d}$*, we have*

$$\lim_{T\to\infty} R_{\text{PCM}}(n, m, \mathbf{w}, \mathbf{d}) = O\left(\frac{1}{n}\right).$$

Proof. Note that $T \to \infty$ simply means $m \to \infty$, and $\frac{d}{2} \log\left(1 + \frac{n}{\hat{w}_{d,i} m}\right) \to 0$ as $m \to \infty$, completing the proof. □

Remark. According to the corollary, the redundancy vanishes as $T \to \infty$ (or equivalently $m \to \infty$). Therefore, for sufficiently large m, significant performance improvement is expected in terms of the number of bits required to describe a sequence x^n.

Definition. We refer to UcompOM as the optimal universal compression with side information in the sense that it achieves the minimum average redundancy given the side information. We further refer to $R_{\text{OM}}(n, m, \mathbf{w}, \mathbf{d})$ as the average redundancy of the optimal universal compression with side information of size m in our problem setup described in Section 5.1, i.e., $T = \frac{m}{n}$ sequences from the mixture source are shared between the encoder and the decoder as side information. As such, we have

$$R_{\mathrm{OM}}(n, m, \mathbf{w}, \mathbf{d}) \triangleq H(X^n|\mathbf{Y}^{n,T}) - H_n(\Delta). \tag{131}$$

Theorem 5.3.6. *In the case of* UcompOM*, we have*

$$R_{\mathrm{OM}}(n, m, \mathbf{w}, \mathbf{d}) = \sum_{d=1}^{d_{max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i} \hat{R}_{d,i}$$

$$+O\left(\frac{1}{\sqrt{T}} + \frac{1}{n}\right) \; a.s.,$$

where $\hat{R}_{d,i}$ *is defined in* (130)*.*

Proof. See Section 5.6 for the proof. □

Remark. Theorem 5.3.6 characterizes the redundancy of the optimal universal compression scheme with side information, which uses a memory of size $m = nT$ (T sequences of size n) in the compression of a new sequence of length n. It is natural to expect that the side information will make the redundancy decrease. The redundancy of the UcompOM decreases when $H(\mathbf{w})$ or roughly K is sufficiently small. Again, $K = 1$, gives $R_{\mathrm{OM}}(n, m, 1, d) = \frac{d}{2} \log\left(1 + \frac{n}{m}\right) + O\left(\frac{1}{n} + \frac{1}{\sqrt{T}}\right)$. Further, it is deduced from Theorem 5.3.6 that $\lim_{T\to\infty} R_{\mathrm{OM}}(n, m, \mathbf{w}, \mathbf{d}) = O\left(\frac{1}{n}\right)$ (regardless of $\mathbf{w}$), i.e., the cost of universality would be negligible given that sufficiently large memory (side information) is available. Thus, the benefits of optimal universal compression with side information would be substantial when $H(\mathbf{w})$ is sufficiently small. On the other hand, when $H(\mathbf{w})$ grows very large, no benefit is obtained from the side information in the universal compression and the performance improvement becomes negligible. This is due to the fact that, in light of Theorem 5.3.1, the compression performance for the known source parameters case is already that of the universal compression.

Corollary 5.3.7. *We have*

$$R_{\mathrm{OM}}(n, m, \mathbf{w}, \mathbf{d}) = R_{\mathrm{PCM}}(n, m, \mathbf{w}, \mathbf{d}) + O\left(\frac{1}{\sqrt{T}} + \frac{1}{n}\right) \; a.s.$$

Proof. The corollary is proved by combining Theorems 5.3.4 and 5.3.6. □

Remark. The corollary has significant implications. It states that the performance of optimal universal compression with side information (UcompOM), which uses a memory of size $m = nT$ (T sequences of size n) in the compression of a new sequence of length n is equal to that of the universal compression with perfectly clustered memory (UcompPCM) up to $O\left(\frac{1}{\sqrt{T}} + \frac{1}{n}\right)$ terms. Hence, when T is sufficiently large, we expect that both have the same performance. This indeed demonstrates that *clustering* is optimal for the universal compression with side information. As such, we pursue the clustering of the side information (i.e., memory) in this work in Section 5.4.

5.4 Parametric Clustering for Mixture Models

In this section, we present the parametric clustering solution for network packets. The K-means algorithm can be used for this purpose provided that proper feature space and distance metrics are selected.

5.4.1 Feature Extraction

Feature extraction deals with extracting simpler descriptions for a large set of data that can accurately describe characteristics of original data. For memoryless source models, the frequency of each alphabet in the sequence defines an empirical probability density distribution vector which also happens to be the sufficient statistics. Although for more sophisticated source models, the empirical probability distribution of the packets is not a sufficient statistics anymore as collisions may occur between different parametric sources in the marginal symbol distribution, the empirical probability distribution would still match for packets from the same source. We choose the vector of the empirical probability distribution as our features and since we work at the byte granularity (i.e., $|\mathcal{A}| = 256$), the feature vector is 255-dimensional. Please note that the chosen feature space is not necessary optimal but simulations confirm that it works well in practice for packets of size 1,500 bytes.

5.4.2 Clustering

As discussed earlier in Section 5.1, we have a side information sequence of packets $\mathbf{y}^{n,T}$ that consists of T packets that originated from a mixture source model. The goal is to classify the packets into K different clusters without knowing K. Intuitively, we can think of a cluster as a group of packets that are close to each other in some space defined by a distance metric when compared with the distances to points outside of the cluster. We choose to use the Euclidean distance metric between any two packets. Please note that Euclidean distance metric is not necessarily the optimal metric for the purpose of clustering of network packets. For each packet $y^n(t)$, we introduce a binary indicator $b_{tk} \in \{0,1\}$ (where $k = 1, \dots, K$) that describes which of the K clusters the packet $y^n(t)$ is assigned to, so that if packet $y^n(t)$ is assigned to cluster k then $b_{tk} = 1$, and $b_{ik} = 0$ for $i \neq k$ [21]. We can define an objective function as

$$J = \sum_{i=1}^{k} \sum_{k=1}^{K} b_{ik} \left|\left| y^n(t) - u_k \right|\right| ,$$

which represents the sum of the distances of each packet to its assigned vector u_k, where u_k is the probability distribution vector of the symbol set. The goal of clustering algorithm is to find values for the $\{b_{ik}\}$ and the $\{u_k\}$ that minimize J.

5.4.3 Classification

Once the clustering of memory is performed, to compress a test packet x^n, we need to classify the packet to one of the K clusters. Then we use the assigned cluster of packets as the side information to compress x^n. The test packet x^n is assigned to the closest cluster by function

$$b = \arg \min_{1 \leq j \leq K} \left|\left| x^n - u_j \right|\right| .$$

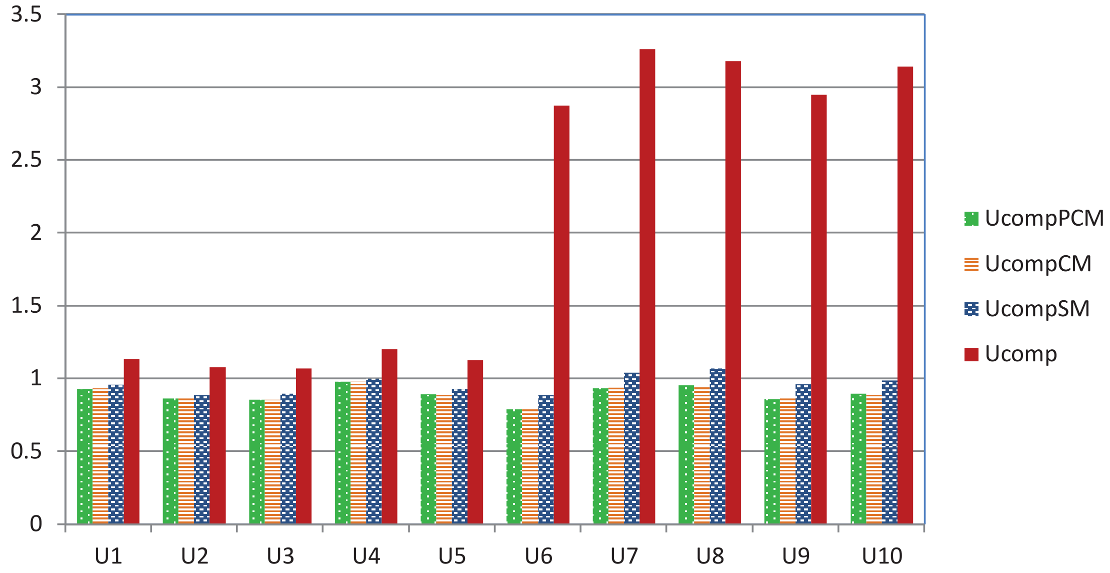

Figure 18: Average compression-rate for a mixture of five memoryless and five first-order Markov sources.

5.5 Simulation Results

The simulations are divided to two parts. In the first part, we generate mixture sources and validate our theoretical findings. Next, we also present results of simulation on real network traffic traces.

5.5.1 Simulations on Man-Made Mixture Models

In order to validate the theoretical results of Section 5.3, we chose to use a mixture of parametric sources as the content-generator for the traffic. In particular, we used a mixture of five memoryless and five first-order Markov sources on 256-ary alphabet ($|\mathcal{A}| = 256$). Consequently for a memoryless Markov source the number of source parameter d is 255, and for a first-order Markov source d is 256×255 which is the number of independent transition probabilities. Further, we assume that each packet is selected uniformly at random from the abovementioned mixture. For short-length sequences, we generate 18,000 packets at random from this source model, where each packet is 1,500 bytes long. Then, we use 200 packets from each source as test packets

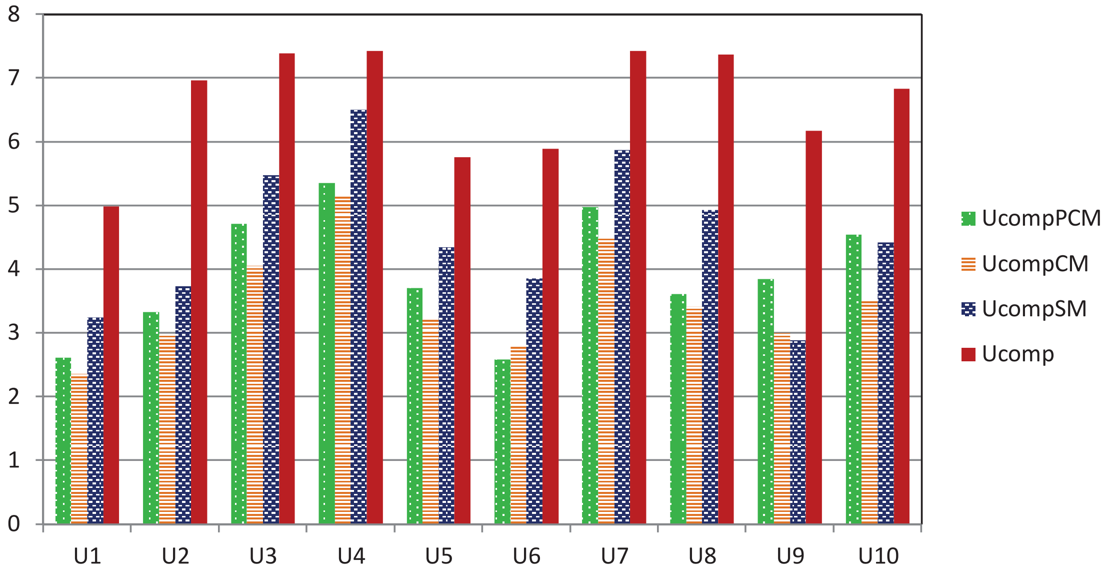

Figure 19: Average compression-rate for the data gathered from the mixture of 10 network users.

for the purpose of evaluation and average out the result.

Figure 18 demonstrates the results of the simulation on man-made data generated from the described mixture source. Users U1 through U5 are memoryless whilst users U6 through U10 are first-order Markov sources. We use lite PAQ-based compression for Ucomp, UcompSM, UcompCM, and UcompPCM. As can be seen, lite PAQ is already doing a poor job when the sequence is from a first-order Markov demonstrating the need for memory-assisted compression. This is also in agreement with the predictions from [16, 52]. We can see that UcompPCM is consistently better than UcompSM and UcompCM as it is the optimal way of classification and clustering, however, UcompPCM is impractical in most scenarios.

5.5.2 Simulations on Real Network Traces

Next, we perform experiments on data gathered from 10 wireless users using real network traces. We chose to mix the data from these 10 users in order to simulate the situation that occurs in an intermediate router that is serving these users. Figure 19 contains the average compression-rate on these data. Please note that we slightly

Table 1: The average compression rate (bits/byte) and average traffic reduction (%) of different compression schemes on the real network traffic traces using lite PAQ.

Scheme	Avg. Comp. Rate	Avg. Traffic Reduc.
Ucomp	6.62	17.2%
UcompSM	4.53	43.4%
UcompPCM	3.93	50.9%
UcompCM	3.50	56.2%

abused the notation and used UcompPCM for compression based on the user from which the data is gathered (and not the *unknown* content-generating source). Here, indeed we do not have access to anything other than the user ID. As can be seen, UcompCM, which is the cluster-based memory-assisted compression presented in this chapter, consistently outperforms all other schemes as data from one user is not necessarily from one source. Table 1 demonstrates the average compression-rate over all the ten users as well the average traffic reduction achieved in this scenario. As can be seen, while lite PAQ (which is one of the very best compression algorithms) only offers 17% traffic reduction on average for the data gathered from these 10 users, by using cluster-based memory-assisted compression more than 50% traffic reduction is achieved. Furthermore, clustering offers more than 5% improvement over the situation where the data from the users are clustered according to the user they are destined to. This confirms that the data destined to a single user are not necessarily from the same content-generating source.

5.6 Technical Analysis

Proof of Theorem 5.2.1. Let D be the random dimension of the source parameter vector. Please note that it is straightforward to show that

$$H(X^n|\Delta) = H(X^n|\Delta, Z, D) + I(X^n; Z, D|\Delta)$$

Further, if Z is known, D is determined, and hence, $H(X^n|\Delta, Z, D) = H(X^n|\Delta, Z)$ which is derived in (123). On the other hand, we have

$$I(X^n; Z, D|\Delta) = I(X^n; D|\Delta) + I(X^n; Z|\Delta, D).$$

Let us first focus on $I(X^n; D|\Delta)$. We have

$$I(X^n; D|\Delta) = H(D|\Delta) - H(D|\Delta, X^n).$$

Please note that $H(D|\Delta)$ is by definition equal to $H(\mathbf{v})$. Further,we can use the maximum likelihood estimator of D using x^n, asymptotically as $n \to \infty$, to consistently estimate D and hence $H(D|\Delta, X^n) = O\left(\frac{1}{n}\right)$ almost surely. Therefore, $I(X^n; D|\Delta) = H(\mathbf{v}) + O\left(\frac{1}{n}\right)$.

Next, we consider $I(X^n; Z|\Delta, D)$. In this case, we have

$$I(X^n; Z|\Delta, D) = \sum_{d=1}^{d_{\max}} I(X^n; Z|\Delta, D = d).$$

In order to analyze, we need to consider two situations. First, let $H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n$. We have

$$I(X^n; Z|\Delta, D = d) = H(Z|\Delta, D = d)$$
$$-H(Z|X^n, \Delta, D = d).$$

Clearly, $H(Z|\Delta, D = d) = H(\hat{\mathbf{w}}_d)$ by definition. Furthermore, if $H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n$, it is straightforward to deduce that $H(Z|X^n, \Delta, D = d) = O\left(\frac{1}{n}\right)$ *a.s.* since the maximum likelihood estimator for the source parameter vector almost surely converges to the true θ, which in turn determines Z. Therefore, if $H(\hat{\mathbf{w}}_d) \prec \frac{d}{2}\log n$, then $I(X^n; Z|\Delta, D = d) = H(\hat{\mathbf{w}}_d) + O\left(\frac{1}{n}\right)$.

To complete the proof of the theorem, we need to show that if $H(\hat{\mathbf{w}}_d) \succ \frac{d}{2}\log n$, we have $I(X^n; Z|\Delta, D = d) = \bar{R}_{n,d} + O\left(\frac{1}{n}\right)$. In this case for any $\epsilon > 0$, there exists a subset of the K_d vectors of the d-dimensional parameter vectors indexed with K'_d, such that

$$(1 - 2\epsilon)\bar{R}_{n,d} < H(\hat{\mathbf{u}}_{\mathbf{d}}) < (1 - \epsilon)\bar{R}_{n,d},$$

where $\hat{\mathbf{u}}_d$ denotes the weight vector of those K'_d parameter vectors renormalized to sum to one. Please note that $\bar{R}_{n,d} \sim \frac{d}{2}\log n$ and hence $H(\hat{\mathbf{u}}_{\mathbf{d}}) \prec \frac{d}{2}\log n$. Therefore, we have $I(X^n; Z|\Delta, D = d) \geq (1-2\epsilon)\bar{R}_{n,d} + O\left(\frac{1}{n}\right)$ almost surely. On the other hand, we also have

$$I(X^n; Z|\Delta, D = d) \leq I(X^n; \theta^{(Z)}|D = d) = \bar{R}_{n,d}.$$

Hence, we deduce that $I(X^n; Z|\Delta, D = d) = \bar{R}_{n,d} + O\left(\frac{1}{n}\right)$ almost surely, completing the proof. □

Proof of Theorem 5.3.1. Let D be the random dimension of the source parameter vector. Please note that it is straightforward to show that

$$H(X^n) = H(X^n|\Delta, Z, D) + I(X^n; \Delta, Z, D) \tag{132}$$

Again, as shown in the proof of Theorem 5.2.1, $H(X^n|\Delta, Z, D) = H(X^n|\Delta, Z)$ as given by (123). So as to proceed, we have

$$I(X^n; \Delta, Z, D) = I(X^n; D) + I(X^n; Z|D) + I(X^n; \Delta|Z, D).$$

Note that $I(X^n; D) = H(\mathbf{v}) + O\left(\frac{1}{n}\right)$ almost surely as $H(D|X^n) = O\left(\frac{1}{n}\right)$ almost surely. Further, $I(X^n; Z|D) = 0$ as X^n does not carry any information about the index of source parameter vector when D is known and no other information is available about the parameter vectors. Please also note that since the parameter vectors are chosen independently, we have $I(X^n; \Delta|Z, D) = I(X^n; \theta^{(Z)}|Z, D)$. Hence,

$$I(X^n; \Delta, Z, D) = I(X^n; \theta^{(Z)}|Z, D) + H(\mathbf{v}) + O\left(\frac{1}{n}\right) \quad a.s.$$

In other words, all the information that X^n carries about the set Δ of the unknown parameter vectors, the index Z, and the dimension D is contained in $I(X^n; \theta^{(Z)}|Z, D)$. On the other hand, as each of the unknown parameter vectors follow Jeffreys' prior, we have $I(X^n; \theta^{(Z)}|Z = z, D = d) = \bar{R}_{n,d}$ [52]. Thus,

$$I(X^n; \theta^{(Z)}|Z, D) = \sum_{d=1}^{d_{\max}} v_d \sum_{i=1}^{K} \hat{w}_{d,i} I(X^n; \theta^{(Z)}|Z = z)$$

$$= \sum_{d=1}^{d_{\max}} v_d \bar{R}_{n,d}. \tag{133}$$

The proof is completed by noting that $R(n, \mathbf{w}, \mathbf{d}) = H(X^n) - H_n(\Delta)$ and substituting $H_n(\Delta)$ from Theorem 5.2.1. □

Proof of Theorem 5.3.3. Recall that UcompSM uses the minimax estimator for the unknown source parameter vector (which is the KT-estimator [47] in the case of memoryless sources) from the T previous sequences by pretending that all the sequences were generated by a single unknown parameter vector. Therefore, if $K = 1$, the minimax estimator using the previous sequences is consistent and indeed achieves the average minimax redundancy. On the other hand, Gallager proved that the average minimax redundancy is equal to the average maximin redundancy [34], which is given by the capacity of the channel between the observed sequence x^n and the unknown source parameter vector θ given the observed sequence $\mathbf{y}^{n,T}$. Formally,

$$R_{\text{SM}}(n, m, 1, d) = \sup_{\omega(\theta)} I(X^n; \theta | \mathbf{Y}^{n,T})|_{\omega(\theta)},$$

where we use the notation $I(X^n; \theta|\mathbf{Y}^{n,T})|_{\omega(\theta)}$ to emphasize that θ follows the prior $\omega(\cdot)$. Apparently, $\omega(\theta) = p_J(\theta)$ (i.e., Jeffreys' prior) serves as a lower limit on the capacity, i.e.,

$$R_{\text{SM}}(n, m, 1, d) \geq I(X^n; \theta | \mathbf{Y}^{n,T})|_{p_J(\theta)}$$

$$= I(X^n, \mathbf{Y}^{n,T}); \theta)|_{p_J(\theta)} - I(\mathbf{Y}^{n,T}); \theta)|_{p_J(\theta)}$$

□

Proof of Theorem 5.3.4. The proof follows the lines of the proof of Theorem 5.3.6 and is omitted for brevity. □

Proof of Theorem 5.3.6. In the case of UcompOM, we have

$$H(X^n|\mathbf{Y}^{n,T}) = H(X^n|\mathbf{Y}^{n,T}, \mathbf{S}, Z) + I(\mathbf{S}, Z; X^n|\mathbf{Y}^{n,T}). \tag{134}$$

On the other hand, we also have

$$H(X^n|\mathbf{Y}^{n,T}\!, \mathbf{S}, Z) = H_n(\Delta, Z) + I(X^n; \theta^{(D,Z)}|\mathbf{Y}^{n,T}\!, \mathbf{S}, Z). \tag{135}$$

The proof of (a) is completed by combining Lemmas 5.6.1 and 5.6.2.

Lemma 5.6.1. *If* $K = O\left(n^{\frac{\check{d}}{2}(1-\epsilon)}\right)$ *for some* $\epsilon > 0$, *then*

$$I(X^n; \theta^{(D,Z)}|\mathbf{Y}^{n,T}, \mathbf{S}, Z) = \hat{R}_i + O(T^{-\frac{1}{2}}),$$

where $\hat{R}_i$ *is defined in* (130).

The proof of Lemma 5.6.1 is carried out by rewriting the LHS as $I(X^n, \mathbf{Y}^{n,T}; \theta^{(D,Z)}|\mathbf{S}, Z) - I(\mathbf{Y}^{n,T}; \theta^{(D,Z)}|\mathbf{S}, Z)$.

Lemma 5.6.2. *If* $K = O\left(n^{\frac{\check{d}}{2}(1-\epsilon)}\right)$ *for some* $\epsilon > 0$, *then*

$$I(\mathbf{S}, Z; X^n|\mathbf{Y}^{n,T}) = H(\mathbf{w}) + O\left(\frac{1}{n} + \frac{1}{T}\right).$$

The proof of Lemma 5.6.2 is carried out by rewriting the LHS as $H(Z|\mathbf{Y}^{n,T}, \mathbf{S}) + H(\mathbf{S}|\mathbf{Y}^{n,T}) - H(\mathbf{S}, Z|\mathbf{Y}^{n,T}, X^n)$ and demonstrating that the last two terms asymptotically vanish. For Part (b), when $w_i = \frac{1}{K}$ and $K = \Omega\left(n^{\frac{d_{\max}}{2}(1+\epsilon)}\right)$ for some $\epsilon > 0$, we have $R(n, \mathbf{w}, \mathbf{d}) = O\left(\frac{1}{n}\right)$ *a.s.* On the other hand, $R_{\mathrm{OM}}(n) \leq R(n, \mathbf{w}, \mathbf{d})$, which completes the proof. □

5.7 Conclusion

In this chapter, the problem of memory-assisted network packet compression for mixture sources was studied from a theoretical point of view. Several different possible schemes for memory-assisted compression of mixture sources were compared. It was proved that cluster-based memory-assisted compression is indeed optimal. A simple clustering algorithm based on K-means clustering was provided for memory-assisted compression of network packets. Our simulation results validated the effectiveness of the clustering for memory-assisted of network packets in practice.

CHAPTER VI

UNIVERSAL COMPRESSION USING ONE-TO-ONE CODES

Thus far, in all the previous chapters, the scope of the network compression via memory has been limited to prefix-free universal codes. The prefix constraint (also known as unique decodability constraint) ensures that a *stream* of data blocks can be uniquely decoded. However, this requirement is too restrictive in many applications, such as the compression of network packets in which IP already marks the beginning and the end of each packet. In such applications, the goal is to uniquely decode *one* block of data. In this scenario, a so called *one-to-one* code without the prefix constraint can still be uniquely decoded if it is strictly lossless (bijective mapping as in Definition 2.1.1).While the average codeword length of prefix-free codes is bounded from below by the entropy, the average codeword length of one-to-one codes can be below the entropy (cf. [2, 43, 50, 82, 83] and the references therein).

Many developments on the fundamental limits of one-to-one codes have taken place very recently. When the source parameter vector is known, the optimal average codeword length for one-to-one codes is known to be upper bounded by entropy. Alon and Orlitsky further derived a lower bound on the average codeword length in [2]. It was shown that the reduction in the average codeword length without the prefix constraint is at most $\log(H(X^n)+1)+\log e$. It was further shown that this bound is attained for geometric distribution. In [82], Szpankowski considered the one-to-one compression of binary memoryless sources and showed that Alon and Orlitsky's bound (which is $\log n + O(1)$ for memoryless sources) is indeed not tight for memoryless sources as the average codeword length of the optimal one-to-one coedes is asymptotically $\frac{1}{2}\log n + O(1)$ below the entropy. In [43], Kontoyiannis and Verdu extended this analysis to general finite-alphabet memoryless sources and proved that $\frac{1}{2}\log n$ holds for all finite-alphabet memoryless sources.

While the performance of one-to-one codes has been investigated extensively for the case of known source parameter vectors, our interest is in the performance of *universal* one-to-one codes for the reasons pointed out in Chapter 1. As has been shown in Chapter 2, universality imposes an inevitable redundancy to the performance of universal compression with prefix constraint. It is desired to know whether or not universal one-to-one codes can perform significantly better than the universal prefix-free codes. Universal one-to-one codes are only developed very recently for memoryless sources by Kosut and Sankar [46], where it was shown that the average redundancy of their proposed universal one-to-one code scales as $\frac{|\mathcal{A}|-3}{2}\log n + O(1)$. This is significantly above the entropy-rate and it is desirable to know whether or not this can be improved. Further, it is desirable to extend these results to the more general class of smooth parametric sources, which contain memoryless and Markov sources as special cases.

In this chapter, the performance of universal one-to-one codes for parametric sources is considered. It is shown that the reduction in the average codeword length due to relaxing the prefix constraint is negligible compared with the overhead associated with the universality of the compression. It is also shown that the code of Kosut and Sankar [46] achieves the optimal second-order term in the redundancy for memoryless sources.

The rest of this chapter is organized as follows. In Section 6.1, the background on one-to-one codes is reviewed. In Section 6.2, our main results on the performance of universal one-to-one codes are presented. The conclusion is given in Section 6.3

6.1 Background on One-to-One Codes

In this section, the existing results on the performance of one-to-one codes are reviewed. Thus far, in all the previous chapters we only considered codes that satisfy Kraft's inequality stated in (2), which is

$$\sum_{x^n \in \mathcal{A}^n} 2^{-l_n(x^n)} \leq 1.$$

As described earlier, Kraft's inequality ensures that a stream of codewords are uniquely decoded to the original sequences. As discussed in detail in Chapter 2, when Kraft's inequality is considered, the minimum average codeword length is achieved using the optimal non-universal length function

$$l_n(x^n) = \log\left(\frac{1}{\mu_\theta(x^n)}\right),$$

and the minimum average codeword length is the entropy. Hence, entropy is a lower bound on the average codeword length , i.e., $\mathbf{E}l_n(x^n) \leq H_n(\theta)$. However, the prefix constraint can be relaxed for the compression of the network packets where the start and end of a sequence are already marked by the header of the packet. This section describes the performance improvement obtained by relaxing Kraft's inequality.

6.1.1 Non-Universal One-to-One Codes

First, we review the non-universal one-to-one codes. Let $l_n^\star(\cdot)$ denote a strictly lossless one-to-one length function. Further, denote $L_n^\star$ as the collection of all one-to-one codes (bijective mappings to binary sequences) on sequences of length n. The following result due to Alon and Orlitsky sets a lower limit on the one-to-one average codeword length.

Theorem 6.1.1. [2]: *Assume that the entropy of the random sequence X^n is equal to $H(X^n)$. Then, the*

$$\mathbf{E}l_n^\star(X^n) \geq H(X^n) - \log(H(X^n) + 1) - \log e. \tag{136}$$

Remark. Theorem 6.1.1 is indeed a very deep result stating that the reduction in the average codeword length associated with a random sequence X^n is at most $\log(H(X^n)+1)+\log e$. Further, Alon and Orlitsky showed that if X^n follows the geometric distribution, the lower limit is attained. Although this provides with a lower bound on the performance, it is desirable to see how tight it is.

When the source statistics are known, we can order all probabilities of the 2^n sequences in a decreasing fashion and then assign a codeword length $\lfloor \log j \rfloor$ to the j-th message sequence. It is straightforward to see that this coding strategy is the optimal one-to-one code but what is perhaps not straightforward is to analyze the average codeword length resulting from this coding strategy. In [82], Szpankowski derived the average codeword length of the non-universal one-to-one codes for binary memoryless sources, recently generalized by Kontoyiannis and Verdu [43] for finite-alphabet memoryless sources as the following.

Theorem 6.1.2. [43, 82]: *In the non-universal one-to-one compression of finite-alphabet memoryless sources, the average codeword length is given by*

$$\mathbf{E}l_n^\star(X^n) = H_n(\theta) - \frac{1}{2}\log n + O(1). \tag{137}$$

Szpankowski tends to call the second-order term the *anti-redundancy* [82], which is the average codeword length reduction below the entropy. Therefore, the anti-redundancy in the non-universal one-to-one compression of finite-alphabet memoryless sources is $\frac{1}{2}\log n + O(1)$ when Kraft's inequality is relaxed.

6.1.2 Universal One-to-One Codes

Thus far, it was shown that for non-universal one-to-one codes the optimal average codeword length is below the entropy. On the other hand, several challenges arise when universal one-to-one codes are concerned. First, the optimal codeword length assignment is no longer obvious. Further, the analysis of a given codeword length assignment is not straightforward.

Let $R^\star_n(l^\star_n, \theta)$ denote the average redundancy of the one-to-one code, which is defined in the usual way as

$$R^\star_n(l^\star_n, \theta) \triangleq \mathbf{E}l^\star_n(X^n) - H_n(\theta). \tag{138}$$

Further, define the one-to-one average maximin redundancy $\underline{R}^\star_n$ as

$$\underline{R}^\star_n = \sup_p \inf_{l^\star_n \in L^\star_n} \int_{\theta \in \Lambda} R^\star_n(l^\star_n, \theta) p(\theta) d\theta, \tag{139}$$

where the supremum is taken over all distributions over the space Λ. Let the one-to-one average minimax redundancy $\bar{R}^\star_n$ be defined as

$$\bar{R}^\star_n = \inf_{l^\star_n \in L^\star_n} \sup_{\theta \in \Lambda} R^\star_n(l^\star_n, \theta). \tag{140}$$

It is straightforward to deduce the following.

Theorem 6.1.3. [34]: *The one-to-one average minimax redundancy is no smaller than the one-to-one average maximin redundancy. That is*

$$\bar{R}^\star_n \geq \underline{R}^\star_n. \tag{141}$$

Remark. According to Theorem 6.1.3, the average minimax redundancy is always at least equal to the average maximin redundancy. Please note that for the case of prefix-free codes it can be shown that they are equivalent [34], while the equivalence would not readily extend to the one-to-one codes.

To the best of our knowledge, the only existing work on universal one-to-one codes is by Kosut and Sankar [46], who proposed a so-called *type-size* coding scheme based on the type of the sequences [27]. The type of sequence x^n is given by

$$t_{x^n}(a) = \frac{|i : x_i = a|}{n} \text{ for } a \in \mathcal{A}. \tag{142}$$

For a type t, let the type class T_t be defined as

$$T_t = \{x^n \in \mathcal{A}^n : t_{x^n} = t\}. \tag{143}$$

Therefore, $|T_t|$ denotes the size of the type class of the type t, i.e., the total number of sequences with type t. Here, we will present a slightly modified version of the type-size code for the purpose of clarity of discussion, which has essentially the same performance. The type-size code essentially sorts the sequences based on the size of the corresponding type classes in a descending order. Therefore, the sequence x^n may appear before y^n only if $|T_{t_{x^n}}| < |T_{t_{y^n}}|$. Then, the rest is performed by assigning a codeword length $\lfloor \log j \rfloor$ to the j-th message sequence. Let l_n^{tsc} denote the length function associated with the type-size code. The performance of the type-size code was analyzed in [46].

Theorem 6.1.4. [46]: *In the universal one-to-one compression of the class of memoryless sources with alphabet size $\mathcal{A}$, for any $\epsilon > 0$, we have*

$$R_n^{\star}(l_n^{tsc}, \theta) \leq (1+\epsilon)\frac{|\mathcal{A}| - 3}{2}\log n + O(1), \tag{144}$$

where $|\mathcal{A}|$ is the size of alphabet.

Remark. According to Theorem 6.1.4, the one-to-one average redundancy for memoryless sources of alphabet size $|\mathcal{A}|$ is asymptotically bounded from above by the expression $\frac{|\mathcal{A}|-3}{2}\log n + O(1)$, which is smaller than $\frac{|\mathcal{A}|-1}{2}\log n + O(1)$ attributed to prefix-free universal codes. However, it remains open to see whether this bound can be further improved and to asses how significant the improvement is.

6.2 Main Results on Universal One-to-One Codes

In this section, our main results on the universal one-to-one compression performance are presented.

Theorem 6.2.1. *Assume that the unknown parameter vector θ follows Jeffreys' prior $p_J(\cdot)$ given in* (11)*, where θ lies in the $|\mathcal{A}|-1$ dimensional simplex of memoryless parameter vectors. Then, type-size code is optimal for the universal one-to-one compression of finite-alphabet memoryless sources. That is*

$$l_n^{tsc} = \arg \inf_{l_n^\star \in L_n^\star} \int_{\theta\in\Lambda} R_n^\star(l_n^\star, \theta)p(\theta)d\theta. \tag{145}$$

Proof. In order to prove the result, we must demonstrate that type-size coding orders the sequences in a descending fashion based based on their probabilities. We have

$$P(x^n) = \int_{\theta\in\Lambda} \mu_\theta(x^n)p_J(\theta)d\theta. \tag{146}$$

On the other hand, since Jeffreys' prior is asymptotically capacity achieving [34, 52], it asymptotically results in equiprobable types. In other words,

$$\mathbf{P}[t_{X^n} = t] \simeq \binom{n+|\mathcal{A}|-1}{n}. \tag{147}$$

where $\binom{n+|\mathcal{A}|-1}{n}$ denotes the total number of type classes, which is a constant with respect to x^n. Hence,

$$\mathbf{P}[X^n = x^n] \simeq \frac{\binom{n+|\mathcal{A}|-1}{n}}{|T_{t_{x^n}}|}. \tag{148}$$

Therefore, by definition of type-size codes, the type-size coding orders the sequences in a descending fashion based on their probabilities, which completes the proof. □

Remark. According to Theorem 6.2.1, the type-size coding is optimal for the universal one-to-one compression of finite-alphabet memoryless sources and the type-size coding is known to achieve a redundancy that is roughly $\frac{|\mathcal{A}|-3}{2}\log n$. However, it remains open to deduce anything about d-dimensional parametric sources. Furthermore, we restricted the analysis to the case where θ follows the capacity achieving Jeffreys' prior. It is desirable to extend the conclusions to cases where θ follows an arbitrary distribution. This can be done by bounding the average minimax redundancy and the average maximin redundancy.

Theorem 6.2.2. *The one-to-one average maximin redundancy for the family $\mathcal{P}_\Lambda^d$ of d-dimensional parametric sources is bounded from below by*

$$\underline{R}_n^\star \geq \frac{d-2}{2}\log\frac{n}{2\pi e} - \log 2\pi e^2 + \int_{\theta\in\Lambda} |\mathcal{I}(\theta)|^{\frac{1}{2}} d\theta + O\left(\frac{1}{\sqrt{n}}\right). \tag{149}$$

Proof. We have

$$H(X^n) = H(X^n|\theta) + I(X^n;\theta) \tag{150}$$

Assuming that θ follows Jeffreys' prior, we can get

$$H(X^n) = \bar{H}_n + \bar{R}_n, \tag{151}$$

where $\bar{R}_n$ is the average minimax redundancy for prefix-free codes given in (9) and $\bar{H}_n$ is given by

$$\bar{H}_n = \int_{\theta\in\Lambda} H_n(\theta)p_J(\theta)d\theta. \tag{152}$$

Hence, we can now use Theorem 6.1.1 to provide a lower bound on $\mathbf{E}l_n^\star(X^n)$. The proof is completed by seeing that $\log \bar{H}_n \leq \log n$ and noting that the average redundancy for the case where θ follows Jeffreys' prior provides a lower limit on the average maximin redundancy. □

Remark. Theorem 6.2.2 basically states that the one-to-one average maximin redundancy is bounded from below by $\underline{R}_n^\star \geq \frac{d-2}{2}\log n + O(1)$. By using Theorem 6.1.3, we can deduce that the bound also holds for the average minimax redundancy, i.e., $\bar{R}_n^\star \geq \frac{d-2}{2}\log n + O(1)$. It is desirable to see how much reduction is offered by universal one-to-one compression compared with the prefix-free universal compression. By assessing the constants in Theorem 6.2.2, it is straightforward to show that the performance improvement is negligible compared with the overhead imposed by universal compression (i.e., the average minimax redundancy). This leads to the conclusion that the universal one-to-one codes are not of much practical interest.

Finally, let us consider the performance of universal one-to-one codes for the case of memoryless sources. For the case of memoryless sources, Theorem 6.2.2 can be translated as

$$\bar{R}_n^\star \geq \frac{|\mathcal{A}|-3}{2} \log n + O(1), \tag{153}$$

which coincides with the performance of type-size coding. Therefore, we have the following.

Theorem 6.2.3. *Type-size coding is minimax (and maximin) optimal for the universal one-to-one compression of memoryless sources.*

6.3 Conclusion

All the previous chapters focused on the performance of universal compression using prefix-free codes. There are numerous applications in which the prefix constraint is indeed unnecessary, as the beginning and end of each block is already determined using some other mechanism. In this chapter, the performance of universal one-to-one codes without prefix constraint was considered. It was proved that the type-size code proposed earlier in the literature is indeed optimal for universal one-to-one compression of memoryless sources. Further, a lower bound on the average minimax redundancy of universal one-to-one codes was derived. Finally, it was also demonstrated that the reduction on the average redundancy by relaxing the prefix constraint is negligible compared with the cost of universality in universal compression of network packets.

CHAPTER VII

CONCLUDING REMARKS

7.1 Summary of Achievements

Correlation elimination is a promising solution to reduce the number of bits associated with the transmission of the ever increasing massive amount of network traffic as the transmission cost comprises a significant fraction of the costs of maintaining such massive data. To this end, this Ph.D. dissertation sets the stage for applying universal compression to this massive data at the packet level just above layer 3 of the network when the intermediate network nodes are enabled with the capability of memorizing the previous traffic. Using the insights gained from these fundamental performance limits, a novel compression-based framework (called *network compression via memory*) is proposed to efficiently eliminate the correlation from network traffic data with superior performance compared with the existing techniques.

First, it is theoretically demonstrated that the well-known prefix-free universal compression algorithms, when used for the correlation elimination of the relatively small network packets, suffer from an inevitable large compression overhead (called redundancy). The same is also confirmed using simulation on real network traffic. Using the insights gained from these fundamental performance limits, a novel compression-based framework (called *network compression via memory*) is proposed to efficiently eliminate the correlation from network traffic data whenever the network nodes (i.e., the encoder and the decoder) are known to be *memory-enabled.* It is shown that network compression via memory provides with superior performance compared with the existing techniques. It is further proved that by choosing the memory size to be sufficiently large, the correlation in the universal compression can be made arbitrarily small. Consequently, memory-assisted universal compression is an effective correlation elimination technique for network traffic data.

Further, the fundamental limits of elimination of the correlation in the spatial dimension are also investigated, i.e., the correlation in the data collected by a common destination from multiple correlated (but spatially separated) sources. It is shown that significant performance improvement may be obtained by considering this correlation motivating the development of practical enocding/decoding systems that can approach the derived limits.

To build a more realistic model for network data, the source model is extended to a mixture of stationary sources. It is shown that in the presence of such a mixture, clustering of the packets to their original models from the mixture is almost surely optimal in terms of the traffic reduction in the context of network compression via memory. Simulation results demonstrate the effectiveness of the proposed approach by matching the expected gains predicted by theory using K-means clustering of the traffic data.

Finally, the network compression via network memory is extended to one-to-one codes without the prefix constraint. It is shown that in the universal compression for one-to-one codes without the prefix constraint at the finite-length the compression overhead is still significant. Furthermore, the impact of memory-assisted compression is analyzed in this setup.

7.2 Future Research Directions

This Ph.D. dissertation presents the first attempt to suppress network traffic data at the packet level using universal compression, and hence, several directions exist for the continuation of this work. In the following, a list of the more important and impactful future research directions are presented.

7.2.1 Compression of Multiple Correlated Distributed Sources

In Chapter 4, a novel correlation model was proposed for two distributed sources with correlated parameter vectors and the fundamental limits of the universal compression in this setup were studied. It is desirable to extend this correlation model beyond two sources to multiple sources. This will also enable the study of network compression with a mixture of correlated sources which is a more realistic model for the content-generating server at the network.

7.2.2 Optimal Clustering for Memory-Assisted Compression

In Chapter 5, it was proven that clustering of the network packets to their original content-generating sources is optimal for achieving the optimal memory-assisted compression performance. Simulation results using K-means clustering algorithm also validated the concept. However, it remains open to provide the optimal feature vector, distance metric, and clustering algorithm for the memory-assisted compression of network data.

7.2.3 Mismatched Memory in Mobile Users

Throughout this Ph.D. dissertation, it was mainly assumed that the memory (side information) is commonly shared between the encoder and the decoder. Although this can be enforced in the backhaul of the network, the majority of the network nodes are becoming mobile users for which sharing a common memory is unrealistic. In Chapter 4, this assumption was relaxed and the impact of memory that is only available to the decoder was considered from a theoretical standpoint. Further, in [68, 69], the impact of mismatched side information between the encoder and the decoder has been considered. In both cases, achievability schemes were provided by covering the space with high-dimensional spheres. Although the theoretical developments are very encouraging about the feasibility of dealing with mismatched memory, the development of a practical yet efficient method to deal with this problem remains a very important open research direction.

7.2.4 Compression of Encrypted Traffic

In this Ph.D. dissertation, the impact of encrypted data was ignored in the memory-assisted compression. Although a significant chunk of the traffic is unencrypted, the network traffic content is moving toward being all encrypted in the future. Encryption aims at generating an encoded sequence that is comprised of independent bits with equal probability of being 0 or 1. Such a sequence is not compressible in principle, and hence, compression and encryption are two conflicting goals. On the other hand, compression of encrypted data is not completely hopeless as there are promising directions for taking encryption into the picture of compression [42]. However, it remains an open problem to develop ideas for memory-assisted universal compression of the encrypted network traffic.

7.2.5 High-Speed Concurrent Compression

This Ph.D. dissertation did not tackle the important problem of compression speed. This will require efficient implementation of scalable memory-assisted compression algorithms that can compress network packets at the routers on the fly. Combining the memory-assisted compression with parallel compression [12, 48] can provide a fast yet efficient way of implementing this, which remains as an open future research direction.

APPENDIX A

PROOFS

A.1 Proof of Lemma 2.3.2

Proof. Since $\mathbf{E}\log\left(\frac{\mu_\theta(X^n)}{\mu_{\phi^\star}(X^n)}\right) = O(1)$, we know that $||\phi^\star(X^n) - \theta|| = O\left(\frac{1}{\sqrt{n}}\right)$. Now, for each $\lambda > 0$, the set of points in the space such that $B(\theta, \Phi^M) > \epsilon$ lie around the midpoints of the lines connecting any $\phi_i \in \Phi^M$ to its nearest neighbors. The volume for each of these regions is $o\left(\frac{1}{n^{\frac{d}{2}}}\right)$ and since $M = O\left(n^{\frac{d}{2}(1-\epsilon)}\right)$, the total probability measure of such $\theta \in \Lambda$ is $o\left(\frac{1}{n^{\frac{d}{2}\epsilon}}\right)$.

□

A.2 Proof of Lemma 2.3.3

Proof. We may use Taylor series to characterize $D_n(\mu_\theta||\mu_{\phi^\circ})$ as a function of θ at $\theta_0 = \phi^\circ$.

$$\frac{1}{n}D_n(\mu_\theta||\mu_{\phi^\circ}) \approx \mathcal{E}_{\phi^\circ}(\theta) + O(||\theta - \phi^\circ||^3), \tag{154}$$

where

$$\mathcal{E}_{\phi^\circ}(\theta) = \frac{\log e}{2}(\theta - \phi^\circ)^T I_n(\phi^\circ)(\theta - \phi^\circ). \tag{155}$$

Since we are interested in points such that $\frac{1}{n}D_n(\mu_\theta||\mu_{\phi^\circ}) = O\left(\frac{1}{n}\right)$, then the error term in (154) uniformly converges to zero with rate $O\left(n^{-\frac{3}{2}}\right)$. Note that $\mathcal{E}_{\phi^\circ}(\theta) \le \delta$ for any $\delta > 0$ defines an ellipsoid around ϕ° in the d-dimensional space of θ. Let $\Delta_d(\phi^\circ, \delta)$ denote the shape characterized by $\frac{1}{n}D_n(\mu_\theta||\mu_{\phi^\circ}) < \delta$. Further, let $V_d(\phi^\circ, \delta)$ be the volume of $\Delta_d(\phi^\circ, \delta)$. We have

$$\begin{aligned} V_d(\phi^\circ, \delta) &= \int_{\theta \in \Delta_d(\phi^\circ, \delta)} d\theta \\ &= \frac{C_d}{|\mathcal{I}_n(\phi^\circ)|^{\frac{1}{2}}} \left(\frac{2\left(\delta + O\left(n^{-\frac{3}{2}}\right)\right)}{\log e} \right)^{\frac{d}{2}} \\ &= \frac{C_d}{|\mathcal{I}_n(\phi^\circ)|^{\frac{1}{2}}} \left(\frac{2\delta}{\log e} \right)^{\frac{d}{2}} \left(1 + \frac{d}{\delta} O\left(n^{-\frac{3}{2}}\right) \right) \\ &= \frac{C_d}{|\mathcal{I}_n(\phi^\circ)|^{\frac{1}{2}}} \left(\frac{2\delta}{\log e} \right)^{\frac{d}{2}} \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right), \end{aligned} \tag{156}$$

where C_d is the volume of the d-dimensional unit ball.

Since θ follows Jeffreys' prior, the probability measure covered by the shape shape $\Delta_d(\phi^\circ, \delta)$ is given by

$$\begin{aligned} \mathbf{P}_\theta \left[\Delta_d(\phi^\circ, \delta) \right] &= \int_{\theta \in \Delta_d(\phi^\circ, \delta)} \left(\frac{|\mathcal{I}_n(\theta)|^{\frac{1}{2}}}{\int |\mathcal{I}_n(\lambda)|^{\frac{1}{2}} d\lambda} \right) d\theta \\ &= \int_{\theta \in \Delta_d(\phi^\circ, \delta)} \left(\frac{\left(|\mathcal{I}_n(\phi^\circ)|^{\frac{1}{2}} + O\left(\frac{1}{\sqrt{n}}\right) \right)}{\int |\mathcal{I}_n(\lambda)|^{\frac{1}{2}} d\lambda} \right) d\theta \\ &= V_d(\phi^\circ, \delta) \left(\frac{|\mathcal{I}_n(\theta)|^{\frac{1}{2}}}{\int |\mathcal{I}_n(\lambda)|^{\frac{1}{2}} d\lambda} \right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right) \\ &= \frac{C_d}{\int |\mathcal{I}_n(\lambda)|^{\frac{1}{2}} d\lambda} \left(\frac{2\delta}{\log e} \right)^{\frac{d}{2}} \left(1 + O\left(\frac{1}{\sqrt{n}}\right) \right). \end{aligned} \tag{157}$$

This completes the proof of the first claim. Please note that although the volume of the shape is a function of the point ϕ° in the parameter space, the probability measure of the shape does not depend on the point ϕ°.

For the second claim, let the event $\mathcal{V}_i$ be the defined as

$$\mathcal{V}_i = \left\{ \omega \in \Omega : \ \frac{1}{n} D_n(\mu_\theta || \mu_{\phi_i}) < \delta \right\}. \tag{158}$$

Note that there are m choices for ϕ_i. For all $1 < i < m$, in the first claim, we found an upper bound on the probability of the event $\mathcal{V}_i$. Thus, using the union bound, we can upper bound the probability of $\bigcup_{i=1}^m \mathcal{V}_i$. Define the following event.

$$\mathcal{W} = \left\{ \omega \in \Omega : \ \min_{\phi_i \in \Phi^m} \frac{1}{n} D_n(\mu_\theta || \mu_{\phi_i}) < \delta \right\}. \tag{159}$$

The second claim is obtained by noting that

$$\mathcal{W} = \bigcup_{i=1}^{m} \mathcal{V}_i. \tag{160}$$

□

A.3 Proof of Lemma 2.3.7

Proof. First, note that

$$R_n(l_n^{2p}, \theta) - R_n(l_n^{n2p}, \theta) = \mathbf{E} \log \left(\frac{1}{A(\phi^\star(X^n))} \right). \tag{161}$$

According to (43), m increases as ϵ decreases until ϵ is minimized and the average minimax redundancy is achieved as in (46). Let $|S_m(\phi^\star)|$ be the number of the sequences whose optimally estimated point (maximum likelihood estimation) is $\phi^\star$. Increasing m results in the increase of the number of the estimate points. Thus, $|S_m(\phi^\star)|$ decreases with m on the average and so does $A(\phi^\star)$. Therefore, we would conclude that $\mathbf{E} \log \left(\frac{1}{A(\phi^\star(X^n))} \right)$ is an increasing function of m. As discussed earlier, we optimized m in order to find the best lower bound on the average redundancy in Theorem 2.3.1. As can be seen in (43), the optimal value of m is decreasing with ϵ. Thus, in order to maximize $\mathbf{E} \log \left(\frac{1}{A(\phi^\star(X^n))} \right)$, we would need to minimize ϵ. As discussed in the proof of Theorem 2.3.4, by minimizing ϵ, we obtain the average minimax redundancy. Therefore, we have

$$\mathbf{E} \log \left(\frac{1}{A(\phi^\star(X^n))} \right) \leq \bar{R}_n^{2p} - \bar{R}_n^{n2p}, \tag{162}$$

Note that the normalized two-part codes achieve the average minimax redundancy, i.e., $\bar{R}_n^{n2p} = \bar{R}_n$. Thus,

$$\bar{R}_n^{2p} - \bar{R}_n^{n2p} = g(d) + O\left(\frac{1}{n}\right). \tag{163}$$

□

A.4 Proof of Lemma 3.3.7

Proof. By applying the definition of $m^\star$, we have

$$\frac{d}{2n}\frac{n}{m}\log e \leq \frac{\delta}{1-\delta}\frac{H_n(\theta)}{n}. \tag{164}$$

By noting that $\log\left(1+\frac{n}{m}\right) \leq \frac{n}{m}\log e$, we have

$$\frac{d}{2n}\log\left(1+\frac{n}{m}\right) \leq \frac{\delta}{1-\delta}\frac{H_n(\theta)}{n}, \tag{165}$$

and hence, the lemma is proved by noting that $\hat{R}_{\mathrm{M}}(n,m) = \frac{d}{2}\log\left(1+\frac{n}{m}\right)$. □

A.5 Proof of Theorem 3.6.5

Proof. It can be shown that the minimax redundancy is equal to the capacity of the channel between the unknown parameter vector θ and the sequence x^n given the sequence y^m (cf. [52] and the references therein). Thus,

$$\begin{aligned}\bar{R}^0_{\mathrm{UcompED}}(n,m) &= \sup_{\omega(\theta)} I(X^n;\theta|Y^m)\\ &= \sup_{\omega(\theta)}\{I(X^n,Y^m;\theta) - I(Y^m;\theta)\}\\ &\geq \{I(X^n,Y^m;\theta) - I(Y^m;\theta)\}|_{\theta\propto\omega_{\mathrm{J}}(\theta)}\\ &= \bar{R}^0_{\mathrm{Ucomp}}(n+m) - \bar{R}^0_{\mathrm{Ucomp}}(m), \end{aligned} \tag{166}$$

where $\omega_{\mathrm{J}}(\theta)$ denotes the Jeffreys' prior defined in (11), and $\bar{R}^0_{\mathrm{Ucomp}}(\cdot)$ is given in Theorem 3.6.1. Further simplification of (166) leads to the desired result in Lemma 3.6.5.

□

A.6 Proof of Lemma 4.1.5

Proof. According to Lemma 4.5.7, for any $\epsilon > 0$ we have

$$\mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)},\epsilon)\right] \lesssim 2Q_d\left(\delta\frac{\mathcal{T}}{2}\right). \tag{167}$$

Also please note that $Q_d(x)$ converges to zero as $x \to \infty$. Hence, $\theta^{(2)}$ converges to $\theta^{(1)}$ in probability as $\mathcal{T} \to \infty$ as claimed. □

A.7 Proof of Lemma 4.5.2

Proof.

$$\begin{aligned} H(X^n|\mathbf{1}_e, \hat{X}^n) &= (1-p_e)H(X^n|\mathbf{1}_e(X^n,)=0, \hat{X}^n) \\ &\quad + p_e H(X^n|\mathbf{1}_e = 1, \hat{X}^n) && (168) \\ &\leq p_e H(X^n). && (169) \end{aligned}$$

The first term in (168) is zero since if $\mathbf{1}_e = 0$, we have $X^n = \hat{X}^n$ and hence $H(X^n|\mathbf{1}_e(X^n,) = 0, \hat{X}^n) = 0$. The inequality in (169) then follows from the fact that $H(X^n|\mathbf{1}_e = 1, \hat{X}^n) \leq H(X^n)$ completing the proof. □

A.8 Proof of Lemma 4.5.6

Proof. Please note that

$$\begin{aligned} &\mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}(\theta,\epsilon)\right] \\ &\qquad \eqsim \mathbf{P}\left[(\hat{\theta}(X^n)-\theta)^T \mathcal{I}(\theta)(\hat{\theta}(X^n)-\theta) > \epsilon^2\right]. && (170) \end{aligned}$$

On other other hand, $n(\hat{\theta}(X^n)-\theta)^T \mathcal{I}(\theta)(\hat{\theta}(X^n)-\theta)$ follows the chi-squared distribution with d degrees of freedom, i.e., χ^2_d. Thus, we can bound the tail by the following:

$$\mathbf{P}\left[(\hat{\theta}(X^n)-\theta)^T \mathcal{I}(\theta)(\hat{\theta}(X^n)-\theta) > \epsilon^2\right] = \frac{\Gamma\left(\frac{d}{2}, \frac{\epsilon^2}{2}n\right)}{\Gamma\left(\frac{d}{2}\right)}. \tag{171}$$

Please also note that we have

$$\lim_{x\to\infty} \frac{\Gamma(s,x)}{x^{s-1}e^{-x}} = 1. \tag{172}$$

Therefore, since $\frac{\epsilon^2}{2}n \to \infty$ as $n \to \infty$, we have

$$\mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}(\theta,\epsilon)\right] \eqsim Q_d\left(\delta\frac{n}{2}\right), \tag{173}$$

where $Q_d(\cdot)$ is defined in (109), which also completes the proof. □

A.9 Proof of Lemma 4.5.7

Proof. In this case, we have

$$\begin{aligned}&\mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \epsilon)\right] \\ &\qquad \approx \mathbf{P}\left[(\theta^{(2)} - \theta^{(1)})^T \mathcal{I}(\theta^{(1)})(\theta^{(2)} - \theta^{(1)}) > \epsilon^2\right]. \qquad (174)\end{aligned}$$

Further, we also have

$$\theta^{(2)} - \theta^{(1)} = (\theta^{(2)} - \hat{\theta}(Z^{2\mathcal{T}})) + (\hat{\theta}(Z^{2\mathcal{T}}) - \theta^{(1)}). \qquad (175)$$

By noting that $(\theta^{(2)} - \hat{\theta}(Z^{2\mathcal{T}}))$ and $(\hat{\theta}(Z^{2\mathcal{T}}) - \theta^{(1)})$ are independent by definition, we see that

$$\begin{aligned}&\mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \epsilon)\right] \\ &\quad \approx \mathbf{P}\Big[(\theta^{(2)} - \hat{\theta}(Z^{2\mathcal{T}}))^T \mathcal{I}(\theta^{(2)})(\theta^{(2)} - \hat{\theta}(Z^{2\mathcal{T}})) \\ &\qquad + (\theta^{(1)} - \hat{\theta}(Z^{2\mathcal{T}}))^T \mathcal{I}(\theta^{(1)})(\theta^{(1)} - \hat{\theta}(Z^{2\mathcal{T}})) > \epsilon^2\Big]. \qquad (176)\end{aligned}$$

Note that we have used the fact that $\mathcal{I}(\theta^{(2)}) \to \mathcal{I}(\theta^{(1)})$ in probability in (176). Hence, by union bound we have

$$\begin{aligned}&\mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \epsilon)\right] \\ &\quad \lesssim \mathbf{P}\left[\hat{\theta}(Z^{2\mathcal{T}}) \notin \mathcal{E}\left(\theta^{(2)}, \frac{\epsilon}{2}\right)\right] + \mathbf{P}\left[\hat{\theta}(Z^{2\mathcal{T}}) \notin \mathcal{E}\left(\theta^{(1)}, \frac{\epsilon}{2}\right)\right]. \qquad (177)\end{aligned}$$

The desired result is obtained by applying Lemma 4.5.6 to (177). □

A.10 Proof of Lemma 4.5.8

Proof. For any $(\delta_1, \delta_2, \delta_3)$ such that $\delta_1 + \delta_2 + \delta_3 = \epsilon$, we have

$$\begin{aligned}&\mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}\left(\hat{\theta}(Y^m), \epsilon\right)\right] \\ &\quad \le \mathbf{P}\Big[\hat{\theta}(X^n) \notin \mathcal{E}(\theta^{(2)}, \delta_1) \\ &\qquad \cup\ \hat{\theta}(Y^m) \notin \mathcal{E}(\theta^{(1)}, \delta_2)\ \cup\ \theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \delta_3)\Big] \qquad (178)\end{aligned}$$

$$
\begin{aligned}
&\leq \Big\{ \mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}(\theta^{(2)}, \delta_1)\right] \\
&\quad + \mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \delta_2)\right] + \mathbf{P}\left[\hat{\theta}(Y^m) \notin \mathcal{E}(\theta^{(1)}, \delta_3)\right] \Big\} \qquad (179)
\end{aligned}
$$

Hence, we can also optimize $(\delta_1, \delta_2, \delta_3)$ to obtain the best bound as follows.

$$
\begin{aligned}
&\mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}\left(\hat{\theta}(Y^m), \epsilon\right)\right] \\
&\leq \min_{\delta_1+\delta_2+\delta_3=\epsilon} \Big\{ \mathbf{P}\left[\hat{\theta}(X^n) \notin \mathcal{E}(\theta^{(2)}, \delta_1)\right] \\
&\quad + \mathbf{P}\left[\theta^{(2)} \notin \mathcal{E}(\theta^{(1)}, \delta_2)\right] + \mathbf{P}\left[\hat{\theta}(Y^m) \notin \mathcal{E}(\theta^{(1)}, \delta_3)\right] \Big\} \qquad (180) \\
&\leq \min_{\delta_1+\delta_2+\delta_3=\epsilon} \Big\{ Q_d\left(\delta_1 \frac{n}{2}\right) + \\
&\qquad 2Q_d\left(\delta_2 \frac{\mathcal{T}}{2}\right) + Q_d\left(\delta_3 \frac{m}{2}\right) \Big\} \qquad (181) \\
&= 4Q_d\left(\frac{\epsilon}{4}\left(\frac{1}{n} + \frac{1}{\mathcal{T}} + \frac{1}{m}\right)\right) \qquad (182)
\end{aligned}
$$

where the inequality in (181) is due to Lemmas 4.5.6 and 4.5.7, and the equality in (182) is obtained by optimizing $(\delta_1, \delta_2, \delta_3)$. □

REFERENCES

[1] *CISCO Virtual Networking Index Forecast*, 2012.

[2] Alon, N. and Orlitsky, A., "A lower bound on the expected length of one-to-one codes," *IEEE Trans. Info. Theory*, vol. 40, pp. 1670–1672, Sept. 1994.

[3] Alon, N. and Orlitsky, A., "Source coding and graph entropies," *IEEE Trans. Info. Theory*, vol. 42, pp. 1329 –1339, Sept. 1996.

[4] Anand, A., Gupta, A., Akella, A., Seshan, S., and Shenker, S., "Packet caches on routers: the implications of universal redundant traffic elimination," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 219–230, 2008.

[5] Anand, A., Muthukrishnan, C., Akella, A., and Ramjee, R., "Redundancy in network traffic: findings and implications," in *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, (New York, NY, USA), pp. 37–48, ACM, 2009.

[6] Anand, A., Sekar, V., and Akella, A., "SmartRE: an architecture for coordinated network-wide redundancy elimination," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 87–98, 2009.

[7] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M., "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, Apr. 2010.

[8] Atteson, K., "The asymptotic redundancy of Bayes rules for Markov chains," *IEEE Trans. Info. Theory*, vol. 45, pp. 2104 –2109, Sept. 1999.

[9] Bannai, E., *Sphere packings, lattices and groups*, vol. 290. Springer, 1999.

[10] Baron, D. and Bresler, Y., "An O(N) semipredictive universal encoder via the BWT," *IEEE Trans. Info. Theory*, vol. 50, pp. 928–937, May 2004.

[11] Baron, D., Bresler, Y., and Mihcak, M. K., "Two-Part Codes with Low Worst-Case Redundancies for Distributed Compression of Bernoulli Sequences," in *37th Annual Conference on Information Sciences and Systems (CISS '03)*, Mar. 2003.

[12] Baron, D., "Fast parallel algorithms for universal lossless source coding," Ph.D. dissertation, Feb. 2003.

[13] Barron, A., Rissanen, J., and Yu, B., "The minimum description length principle in coding and modeling," *IEEE Trans. Info. Theory*, vol. 44, pp. 2743 –2760, Oct. 1998.

[14] Barron, A. R. and Cover, T. M., "Minimum complexity density estimation," *IEEE Trans. Info. Theory*, vol. 37, pp. 1034–1054, Jul. 1991.

[15] Beirami, A. and Fekri, F., "On the finite-length performance of universal coding for k-ary memoryless sources," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 740 –744, Sept. 2010.

[16] Beirami, A. and Fekri, F., "Results on the redundancy of universal compression for finite-length sequences," in *2011 IEEE International Symposium on Information Theory (ISIT '11)*, pp. 1604–1608, Jul. 2011.

[17] Beirami, A. and Fekri, F., "On lossless universal compression of distributed identical sources," in *2012 IEEE International Symposium on Information Theory (ISIT '12)*, pp. 561–565, Jul. 2012.

[18] Beirami, A., Sardari, M., and Fekri, F., "Results on the fundamental gain of memory-assisted universal source coding," in *2012 IEEE International Symposium on Information Theory (ISIT '12)*, pp. 1087–1091, Jul. 2012.

[19] Beirami, A., Sardari, M., and Fekri, F., "Results on the optimal memory-assisted universal compression performance for mixture sources," in *51st Annual Allerton Conference*, pp. 890–895, Oct. 2013.

[20] Berger, T., Zhang, Z., and Viswanathan, H., "The CEO problem," *IEEE Trans. Info. Theory*, vol. 42, pp. 887 –902, May 1996.

[21] Bishop, C. M., *Pattern recognition and machine learning.* Springer, 2006.

[22] Chandrasekhar, V. and Andrews, J., "Uplink capacity and interference avoidance for two-tier femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 8, pp. 3498 –3509, Jul. 2009.

[23] Chandrasekhar, V., Andrews, J., and Gatherer, A., "Femtocell networks: a survey," *IEEE Communications Magazine*, vol. 46, pp. 59 –67, Sept. 2008.

[24] Clarke, B. and Barron, A., "Information-theoretic asymptotics of Bayes methods," *IEEE Trans. Info. Theory*, vol. 36, pp. 453 –471, May 1990.

[25] Coxeter, H., Few, L., and Rogers, C., "Covering space with equal spheres," *Mathematika*, vol. 6, no. 02, pp. 147–157, 1959.

[26] Csiszár, I. and Talata, Z., "Context tree estimation for not necessarily finite memory processes, via BIC and MDL," *IEEE Trans. Info. Theory*, vol. 52, pp. 1007 –1016, Mar. 2006.

[27] Csiszár, I., "The method of types," *IEEE Trans. Info. Theory*, vol. 44, pp. 2505–2523, Oct. 1998.

[28] Davisson, L., “Universal noiseless coding,” *IEEE Trans. Info. Theory*, vol. 19, pp. 783 – 795, Nov. 1973.

[29] Davisson, L. and Leon-Garcia, A., “A source matching approach to finding minimax codes,” *IEEE Trans. Info. Theory*, vol. 26, pp. 166 – 174, Mar. 1980.

[30] Drmota, M. and Szpankowski, W., “Precise minimax redundancy and regret,” *IEEE Trans. Info. Theory*, vol. 50, pp. 2686–2707, Nov. 2004.

[31] Effros, M., Visweswariah, K., Kulkarni, S., and Verdu, S., “Universal lossless source coding with the Burrows Wheeler transform ,” *IEEE Trans. Info. Theory*, vol. 48, pp. 1061–1081, May 2002.

[32] Feder, M. and Merhav, N., “Hierarchical universal coding,” *IEEE Trans. Info. Theory*, vol. 42, pp. 1354 –1364, Sept. 1996.

[33] Feder, M., Merhav, N., and Gutman, M., “Universal prediction of individual sequences,” *IEEE Trans. Info. Theory*, vol. 38, pp. 1258 –1270, Jul. 1992.

[34] Gallager, R. G., “Source coding with side information and universal coding,” *unpublished.*

[35] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., “Distributed video coding,” *Proceedings of the IEEE*, vol. 93, pp. 71–83, Jan. 2005.

[36] Greenberg, A., Hamilton, J., Maltz, D. A., and Patel, P., “The cost of a cloud: research problems in data center networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 68–73, Dec. 2008.

[37] Grunwald, P. D., *The Minimum Description Length Principle.* The MIT Press, 2007.

[38] Huffman, D. A., “A Method for the Construction of Minimum-Redundancy Codes,” *Proceedings of the I.R.E.*, pp. 1098–1102, Sept. 1952.

[39] Iyer, S., Rowstron, A., and Druschel, P., “Squirrel: a decentralized peer-to-peer web cache,” in *PODC ’02: Proceedings of the twenty-first annual symposium on Principles of distributed computing*, (New York, NY, USA), pp. 213–222, ACM, 2002.

[40] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L., “Networking named content,” in *5th ACM intl. conf. on Emerging networking experiments and technologies (CoNEXT ’09)*, pp. 1–12, 2009.

[41] Jacquet, P. and Szpankowski, W., “Markov types and minimax redundancy for Markov sources,” *IEEE Trans. Info. Theory*, vol. 50, pp. 1393 – 1402, Jul. 2004.

[42] Johnson, M., Ishwar, P., Prabhakaran, V., Schonberg, D., and Ramchandran, K., "On compressing encrypted data," *IEEE Trans. Signal Process.*, Oct.

[43] Kontoyiannis, I. and Verdu, S., "Optimal lossless data compression: Non-asymptotics and asymptotics," *IEEE Trans. Info. Theory*, vol. 60, pp. 777–795, Feb. 2014.

[44] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I., "A data-oriented (and beyond) network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 181–192, Aug. 2007.

[45] Korodi, G., Rissanen, J., and Tabus, I., "Lossless data compression using optimal tree machines," in *2005 Data Compression Conference (DCC '2005)*, pp. 348 – 357, Mar. 2005.

[46] Kosut, O. and Sankar, L., "Universal fixed-to-variable source coding in the finite blocklength regime," in *2013 IEEE International Symposium on Information Theory Proceedings (ISIT '13)*, pp. 649–653, Jul. 2013.

[47] Krichevsky, R. E. and Trofimov, V. K., "The performance of universal encoding," *IEEE Trans. Info. Theory*, vol. 27, no. 2, pp. 199–207, 1981.

[48] Krishnan, N., Baron, D., and Mihcak, M. K., "A parallel two-pass MDL context tree algorithm for universal source coding," in *2014 IEEE International Symposium on Information Theory Proceedings (ISIT '14)*, Jul. 2014.

[49] Langdon Jr., G. G., "An Introduction to Arithmetic Coding," *IBM J. Res. Develop.*, vol. 28, pp. 135–149, Mar. 1984.

[50] Leung-Yan-Cheong, S. and Cover, T., "Some equivalences between shannon entropy and kolmogorov complexity," *IEEE Trans. Info. Theory*, vol. 24, pp. 331–338, May 1978.

[51] Martin, A., Seroussi, G., and Weinberger, M., "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Info. Theory*, vol. 50, pp. 1442 – 1468, Jul. 2004.

[52] Merhav, N. and Feder, M., "A strong version of the redundancy-capacity theorem of universal coding," *IEEE Trans. Info. Theory*, vol. 41, pp. 714 –722, May 1995.

[53] Mittelmann, H. D. and Vallentin, F., "High-accuracy semidefinite programming bounds for kissing numbers," *Experimental Mathematics*, vol. 19, no. 2, pp. 175–179, 2010.

[54] Oohama, Y., "Gaussian multiterminal source coding," *IEEE Trans. Info. Theory*, vol. 43, pp. 1912 –1923, Nov. 1997.

[55] Oohama, Y., "The rate-distortion function for the quadratic gaussian ceo problem," *IEEE Trans. Info. Theory*, vol. 44, pp. 1057 –1070, May 1998.

[56] Pradhan, S. and Ramchandran, K., "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Info. Theory*, vol. 49, pp. 626 – 643, Mar. 2003.

[57] Qiu, D. and Srikant, R., "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 367–378, ACM, 2004.

[58] Raghavan, B., Vishwanath, K., Ramabhadran, S., Yocum, K., and Snoeren, A. C., "Cloud control with distributed rate limiting," in *SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, (New York, NY, USA), pp. 337–348, ACM, 2007.

[59] Rissanen, J., "Universal coding, information, prediction, and estimation," *IEEE Trans. Info. Theory*, vol. 30, pp. 629 – 636, Jul. 1984.

[60] Rissanen, J., "Complexity of strings in the class of Markov sources," *IEEE Trans. Info. Theory*, vol. 32, pp. 526–532, Jul. 1986.

[61] Rissanen, J., "Stochastic complexity and modeling," *Annals of Statistics*, vol. 14, no. 3, pp. 1080–1100, 1986.

[62] Rissanen, J., "Strong optimality of the normalized ML models as universal codes and information in data," *IEEE Trans. Info. Theory*, vol. 47, pp. 1712 –1717, Jul. 2001.

[63] Rissanen, J. and Langdon, G., J., "Universal modeling and coding," *IEEE Trans. Info. Theory*, vol. 27, pp. 12 – 23, Jan. 1981.

[64] Rissanen, J., "Fisher information and stochastic complexity," *IEEE Trans. Info. Theory*, vol. 42, pp. 40 –47, Jan. 1996.

[65] Sanadhya, S., Sivakumar, R., Kim, K.-H., Congdon, P., Lakshmanan, S., and Singh, J. P., "Asymmetric caching: improved network deduplication for mobile devices," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, (New York, NY, USA), pp. 161–172, ACM, 2012.

[66] Sardari, M., Beirami, A., and Fekri, F., "On the network-wide gain of memory-assisted source coding," in *2011 IEEE Information Theory Workshop (ITW '11)*, pp. 476–480, Oct. 2011.

[67] Sardari, M., Beirami, A., and Fekri, F., “Memory-assisted universal compression of network flows,” in *2012 International Conference on Computer Communications (INFOCOM ’12)*, pp. 91–99, Mar. 2012.

[68] Sardari, M., Beirami, A., and Fekri, F., “Wireless network compression: Code design and trade offs,” in *Information Theory and Applications Workshop (ITA)*, Feb. 2013.

[69] Sardari, M., Beirami, A., and Fekri, F., “Mismatched side information in wireless network compression via overhearing helpers,” in *2014 IEEE International Symposium on Information Theory (ISIT ’14)*, Jul. 2014.

[70] Sardari, M., Beirami, A., Zou, J., and Fekri, F., “Content-aware network data compression using joint memorization and clustering,” in *2013 IEEE Conference on Computer Networks (INFOCOM 2013)*, Apr. 2013.

[71] Sartipi, M. and Fekri, F., “Distributed source coding using short to moderate length rate-compatible LDPC codes: the entire Slepian-Wolf rate region,” *IEEE Trans. Commun.*, vol. 56, pp. 400–411, Mar. 2008.

[72] Savari, S., “Redundancy of the Lempel-Ziv incremental parsing rule,” *IEEE Trans. Info. Theory*, vol. 43, pp. 9–21, Jan. 1997.

[73] Shamir, G. I., Tjalkens, T. J., and Willems, F. M. J., “Low-complexity sequential probability estimation and universal compression for binary sequences with constrained distributions,” in *IEEE Intl. Symp. Info. Theory (ISIT)*, (Toronto, Canada), pp. 995–999, 2008.

[74] Shannon, C. E., “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Jul., Oct. 1948.

[75] Shen, S.-H., Gember, A., Anand, A., and Akella, A., “REfactor-ing content overhearing to improve wireless performance,” in *Proc. of the 17th ACM annual international conference on Mobile computing and networking (MobiCom ’11)*, pp. 217–228, 2011.

[76] Shen, S.-H., Gember, A., Anand, A., and Akella, A., “Refactor-ing content overhearing to improve wireless performance,” in *Mobicom*, (Las Vegas, Nevada, US), 2011.

[77] Shields, P. C., “Universal redundancy rates do not exist,” *IEEE Trans. Info. Theory*, vol. 39, pp. 520–524, Mar. 1993.

[78] Shtarkov, Y., “Universal sequential coding of single messages,” *Problems of Information Transmission*, vol. 23, pp. 175–186, 1987.

[79] Slepian, D. and Wolf, J. K., “Noiseless coding of correlated information sources,” *IEEE Trans. Info. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[80] Spring, N. and Wetherall, D., "A protocol-independent technique for eliminating redundant network traffic," in *ACM SIGCOMM*, 2000.

[81] Szpankowski, W., "Asymptotic average redundancy of Huffman (and other) block codes ," *IEEE Trans. Info. Theory*, vol. 46, pp. 2434–2443, Nov. 2000.

[82] Szpankowski, W., "A one-to-one code and its anti-redundancy," *IEEE Trans. Info. Theory*, vol. 54, pp. 4762–4766, Oct. 2008.

[83] Szpankowski, W. and Verdu, S., "Minimum expected length of fixed-to-variable lossless compression without prefix constraints," *IEEE Trans. Info. Theory*, vol. 57, pp. 4017–4025, Jul. 2011.

[84] Szpankowski, W., " Average Redundancy for Known Sources: Ubiquitous Trees in Source Coding ," *DMTCS Proceedings*, vol. 0, no. 1, 2008.

[85] Weinberger, M., Merhav, N., and Feder, M., "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Info. Theory*, vol. 40, pp. 384–396, Mar. 1994.

[86] Weinberger, M., Rissanen, J., and Feder, M., "A universal finite memory source," *IEEE Trans. Info. Theory*, vol. 41, pp. 643 –652, May 1995.

[87] Willems, F., "The context-tree weighting method: extensions," *IEEE Trans. Info. Theory*, vol. 44, pp. 792–798, Mar. 1998.

[88] Willems, F., Shtarkov, Y., and Tjalkens, T., "The context-tree weighting method: basic properties," *IEEE Trans. Info. Theory*, vol. 41, pp. 653–664, May 1995.

[89] Wyner, A. and Ziv, J., "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Info. Theory*, vol. 22, pp. 1–10, Jan. 1976.

[90] Xie, Q. and Barron, A., "Minimax redundancy for the class of memoryless sources," *IEEE Trans. Info. Theory*, vol. 43, pp. 646 –657, Mar. 1997.

[91] Zhuang, Z., Chang, T.-Y., Sivakumar, R., and Velayutham, A., "Application-aware acceleration for wireless data networks: Design elements and prototype implementation," *IEEE Trans. Mobile Comput.*, vol. 8, pp. 1280–1295, Sept. 2009.

[92] Zhuang, Z., Tsao, C.-L., and Sivakumar, R., "Curing the amnesia: Network memory for the Internet," Tech. Report, 2009.

[93] Ziv, J., "A universal prediction lemma and applications to universal data compression and prediction," *IEEE Trans. Info. Theory*, vol. 47, pp. 1528 –1532, May 2001.

[94] Ziv, J. and Lempel, A., "A universal algorithm for sequential data compression," *IEEE Trans. Info. Theory*, vol. 23, pp. 337–343, May 1977.

[95] Ziv, J. and Lempel, A., "Compression of individual sequences via variable-rate coding," *IEEE Trans. Info. Theory*, vol. 24, pp. 530–536, Sept. 1978.