On the Compression of Unknown Sources

A DISSERTATION SUBMITTED

TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAI'I AT MĀNOA IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN
ELECTRICAL ENGINEERING

By
Maryam Hosseini

Dissertation Committee :
Narayana Prasad Santhanam, Chairperson
Anthony Kuh
Anders Høst-Madsen
June Zhang
Yuriy Mileyko

To my parents for their never ending support
and to Seyyed Abolhasan, my beloved husband

Acknowledgements

Many people have contributed to making this thesis possible. First of all, I would like to thank my advisor, Professor Narayana Prasad Santhanam for supporting me during these years. Prasad has always been helpful and encouraging and gave me the freedom to explore new directions while providing me with valuable feedback, insights, and advice. I feel deeply thankful to him for teaching me to be rigorous and precise.

I wish to thank Professor Anthony Kuh, Professor Anders Host-Madsen, Professor June Zhang, Professor Yury Mileyko and Professor Aleksander Kavcic for accepting to be members of my committee and providing invaluable comments and advice. During these years of Ph.D. study, I had the opportunity to take different courses with Professor Host-Madsen and learned random processes, information theory, and detection and estimation from him. I would also like to thank the college of engineering staff, Lian, Joyce, Gail, Arynn, and Jamie for all the helps they have provided during these years.

I am deeply grateful to Professor Vijay Gupta for providing me an excellent visiting opportunity at the University of Notre Dame. During my visit there, I was also very lucky to work with Professor Takashi Tanaka. I spent a wonderful summer at South Bend.

I want to thank my fellow graduate students in the program, Meysam and Ramezan from whom not only I learned a lot about research but also they made my Ph.D. years fun and joyful. During the past six years, I spent most of my time at our lab at POST 325. I have to thank my labmates Kevin, Changlong, Ian, Charles, Mojataba, Philip, Navid, and Elyas. My time spent in Hawaii was enjoyable thanks to my amazing friends Amanda, Sara, Maryam, Susan, Reza, and Ehsan.

My journey to pursue Ph.D. abroad was only possible because of the support and encouragement of my lovely parents. I'm always indebted to them for their love, support, and sacrifices in every step of my life. I truly owe everything to them and have no words to thank them. My deepest thanks go to my siblings, my best friends, Somayye, Homa, and Amirhossein, although they were thousands of miles away from me, their love and support were always with me.

I have to thank my beloved husband, Seyyed, who has been with me all these years. I'm sure I couldn't finish my Ph.D. without his encouragement and patience. I can not thank him enough for all the incredible support he has provided. He was always around at times I thought that it is impossible to continue on this path.

The last year of Ph.D. was a very different one with my new role as a mother. I experienced a new world, a new type of joy, a new set of feelings, and definitely a new level of tiredness. I want to thank my sweetheart, Arman who made these experiences possible and added endless joy to my life. I'm so lucky and thankful to have him.

Abstract

Usually, data is considered as the outcome of probability sources and to get insight from data, we need to know more about its underlying distribution. Although it is very helpful to know the precise characterization of the source, most of the time such information is not available. However, we usually know that the underlying distribution is not completely arbitrary and belongs to a general class of models, such as the class of *i.i.d.* or Markov distributions. While universal compression of finite support distributions has been well studied, we look into more involved classes—in particular, distributions over countable supports, as well as the relations between compression and estimation in Markov setups without mixing assumptions. In the first part of the dissertation, we investigate "compressibility" of a class of distributions. The exact identity of each distribution in the class is unknown, so we aim to find a universal encoder to compress all distributions in the class. But since the universal encoder does not match exactly to the underlying distribution, the average number of bits we use is higher, and the excess bits used over the entropy is the redundancy. We study the redundancy of universal encodings of strings generated by independent identically distributed (i.i.d.) sampling from a set of distributions over a countable support. We show that universal compression of length-n i.i.d. sequences is characterized by how well the tails of distributions in the collection can be universally described, and we formalize the later as the tail-redundancy of the collection. We show that per-symbol redundancy converges to tail redundancy asymptotically and therefore characterize a necessary and sufficient condition for a collection of distributions to be "strongly compressible". We also consider the redundancy of universally compressing strings generated by a binary Markov source without any bound on the memory. We prove asymptotically matching (in order) upper and lower bounds on the redundancy.

Apart from the abstract analysis of a collection of unknown distributions, we adapt and implement an algorithm to compress the data obtained from an unknown source. Compression can be lossless or lossy. For lossless compression, Lempel and Ziv proposed a universal implementable algorithm and prove that their algorithm achieves the theoretical bound asymptotically. However, many applications can tolerate such amount of distortion which may allow for additional compression. We adapt Codelet parsing, a lossy Lempel-Ziv type algorithm. It sequentially parses a sequence to phrases which we call sourcelet and maps them to codelet in the dictionary. We develop concept "strong match" and use Cycle Lemma to make sure that strong match method does not remove most of the possible matches from the tree. We study different properties of this dictionary and monitor how the leaves of the dictionary evolve.

In the last chapter of the dissertation, we use rate-distortion theory to formulate a problem in cyber-physical systems. Consider a process being controlled remotely by a controller. Let an attacker have access to the communication channel so that she is able to replace the signal transmitted by the controller with any signal she wishes. The attacker wishes to degrade the control performance maximally without being detected. The controller wishes to detect the presence of the attacker by watermarking signaling information in the control input without degrading the control performance. We show that in the one-step version of the problem, if the watermark is a Gaussian distributed random variable, then the maximal performance degradation for any given level of stealthiness for the attacker is achieved when the attacker replaces the control input with the realization of a Gaussian random variable. Conversely, we show the watermark signal that minimizes the stealthiness of a Gaussian attacker is also Gaussian.

Contents

1

Introduction

1.1 Universal Compression

Shannon's fundamental paper on compression assures us that any data obtained from distribution p can be compressed with no more bits than its entropy. In fact, if p is known, we can use Huffman coding to achieve this theoretical lower bound. But in most of the applications, p is unknown and the only information we have is that it belongs to a specific collection of distributions, for example, class of *i.i.d.* or Markov sources. In this case, we need to find a universal encoder that can compress all sources in the collection relatively good at the same time.

To characterize the performance of such universal encoder several different techniques has been developed. Let $\mathcal{P}$ be a collection of distributions over a countable support $\mathcal{X}$, the most stringent metric is the *worst-case* formulation that tries to find a universal probability law q over $\mathcal{X}$ minimizing

$$\sup_{p\in\mathcal{P}}\sup_{x\in\mathcal{X}}\log\frac{p(x)}{q(x)}.$$

The *average-case* formulations try to find a law q minimizing

$$\sup_{p\in\mathcal{P}}\sum_{x\in\mathcal{X}}p(x)\log\frac{p(x)}{q(x)}.$$

A weaker notion also proposed by Kieffer [1] where for all $p\in\mathcal{P}$, the encoder q minimizes

$$\sum_{x\in\mathcal{X}}p(x)\log\frac{p(x)}{q(x)}.$$

Starting from [2], the case where $\mathcal{X}$ is either a finite set of size k, or length-n sequences drawn from a finite set k, and $\mathcal{P}$ is a collection of *i.i.d.* or Markov sources have been studied extensively. A cursory set of these papers include [3–5] for compression of *i.i.d.* sequences of sequences drawn from $k-$sized alphabets, [6, 7] for context tree sources, as well as extensive work involving renewal processes [8, 9], finite state sources [10], etc.

However, the finite alphabet assumption is very restrictive. Although usually in classical information theory and statistics sample size is much higher than the alphabet size, this assumption does not hold for a lot of problems like text classification, language modeling, and DNA microarray analysis. For instance, language models for speech recognition estimate distributions over English words using text samples much smaller than the English vocabulary.

Analyzing large alphabet problems is not an easy extension of finite alphabet ones. In fact, removing the assumption of the finiteness of alphabet changes some fundamental behaviors. For instance, while in the finite alphabet case the order of worst-case redundancy and average case redundancy are same, in large alphabet it is possible to construct an example that worst case redundancy is infinite but average case redundancy is finite.

We first present some preliminaries and basic definitions in universal compression in chapter 2 and briefly review existing works on patterns, weak compression, worst case redundancy and metric entropy. Unlike the weak compression and worst case formulation that finite single letter redundancy (worst case redundancy, respectively) implies vanishing per symbol redundancy (the average case redundancy over the length of the sequence), we show that it is not true in average case redundancy. We construct an example that single letter is finite but per-symbol redundancy does not go to zero as n (length of the sequence) goes to infinity. We study average case redundancy over the countable alphabet and its connection to tightness in chapter 3 and obtain conditions that characterize in which case per-symbol redundancy goes to zero. Although this condition is sufficient but is not necessary. However, this characterization gives us the insight to develop concept *tail redundancy* which is defined as

$$\mathcal{T}(\mathcal{P}) = \lim_{m\to\infty} \inf_{q} \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x) \log \frac{p(x)}{q(x)},$$

where the infimum is over all distributions q over $\mathbb{N}$.

In chapter 4, we first study properties of tail redundancy. We show that the limit in the definition of tail redundancy exists and there is always an encoder that achieves the tail redundancy. Furthermore, we prove that tail redundancy is always positive. Later, we show that universal compression of length-n i.i.d. sequences is characterized by how well the tails of distributions in the collection can be universally described, and we prove that zero tail redundancy is a necessary and sufficient condition for vanishing per symbol redundancy.

In chapter 5, we study redundancy of Markov processes without any assumption on mixing and no hard constraint on the memory. Although we look at binary Markov sources, since we do not impose any hard constraint on the memory, the state space can be large which makes the problem very similar to large alphabet finite memory classes. Normally, memory is a natural hierarchy to consider for Markov sources and if the memory is bounded by m

then redundancy is $2^{m-1}\log n$. But memory it is not a reasonable ordering parameter from estimation point of view. A long memory source may be easier to estimate than a short memory source that doesn't search state space efficiently. We look at Markov sources with continuity condition. Continuity condition implies that the influence of prior symbols dies down as we look further into the past. In the absence of an upper bound on the memory, the continuity condition implies that $p(X_0|X_{-m}^{-1})$ gets closer to the true probability $p(X_0|X_{-\infty}^{-1})$ as m increases, rather than vary around arbitrarily.

We then answer two questions. First, what is the Redundancy of a collection of Markov sources without any mixing assumption and no bound on the memory of the collection? Second, which sources contribute more to the redundancy? We obtain matching (in order) lower and upper bound on the memory of Markov sources and we show that in compressing unbounded Markov sources the primary contribution comes from sources whose state probabilities are not near 0 or 1. To obtain the lower bound we use redundancy capacity theorem. Although it is the common technique to obtain the lower bound, it is not an easy extension when there is no hard bound on memory. In fact, the crucial part is bounding the estimation error. To obtain the upper bound we use the fact that the probability of the collection can be bounded by an aggregated model with bounded memory with high probability.

1.2 Lossy Compression

In the second part of the dissertation, we study a practical problem. Compression can be lossless or lossy. For lossless compression, Lempel and Ziv proposed a universal implementable algorithm and prove that it achieves the theoretical bound asymptotically. Currently, LZ-based algorithms are in use different file formats. There are many applications that some amount of distortion is tolerable to reduce the number of bits required to code. Let x^n be a sequence that we want to represent with another sequence $\hat{x}^n$ that needs less bit to describe. Given a specific distortion level d, what is the minimum compression rate that is achievable? Shannon formulated this problem as rate-distortion problem and showed that *rate-distortion* *function* is a lower bound for the compression rate. In fact, he showed that the optimal $\hat{x}^n$ has *optimal reproduction type.* For example, assume $x^n \sim \mathcal{B}(p)$, then for a given distortion level d, the optimal reproduction type is $\frac{p-d}{1-2d}$. Therefore, type (number of one over length of the sequence) of an optimal representation $\hat{x}^n$ must be equal to $\frac{p-d}{1-2d}$.

Extending Lempel-Ziv algorithm for lossy compression is not an easy task and in fact constructing an online implementable optimal algorithm in lossy compression is still an open problem. Lossless Lempel-Ziv constructs a tree (also known as a dictionary) from the sequence x^n, but adapting this tree when there is a room for distortion is a difficult problem. In fact, in presence of distortion the search space is large so that any naive optimal algorithm will be computationally infeasible.

We adapt Codelet parsing from [60], a polynomial time lossy Lempel-Ziv type algorithm. In the heart of the Codelet Parsing is the concept of *strong match* [11]. We say that two sequences x^n and $\hat{x}^n$ matches if

$$1/j \sum_{i=1}^{j} d_H(x_i, \hat{x}_i) < d, 1 \leq j \leq n, d \leq 0.5.$$

The development of strong match inspired by *Cycle Lemma.* Cycle lemma is a very profound but simple result that rediscovered in literature multiple times. To explain it, let us first define $k-$dominating sequence. A sequence $p_1p_2...p_{m+n}$ of zeros and ones is $k-$dominating if number of zeros in every subsequence $p_1p_2...p_i$, $1 \leq i \leq m+n$ is greater than k times of number ones. For example, sequence “00010010” is 2-dominating, “00101001” is 1-dominating and “10000000” and “00110001” are not even 1-dominating. Cycle lemma states that for any sequence containing m zeros and n ones where $m \geq kn$, number of cyclic permutation of which are $k-$dominating is $m - kn$. Codelet parsing considers strong match instead of convention notion of matching which is $1/n \sum_{i=1}^{n} d_H(x_i, \hat{x}_i) \leq d$. Cycle lemma guarantees that with confining to strong match we are not loosing too many matches and at the same the computational complexity greatly reduces.

Even with considering the strong match, there may be more than one possible match. Therefore, we adapt a different variation of Codelet parsing. In chapter 6, we first explain backbone of our algorithm and some naive variation that choose the *"first match"* and *"longest match"* and propose simulation results of compression rate and time complexity for different value of p and d for data dawn from $\mathcal{B}(p)$ distribution. Since the shape of the tree that evolves and type of its leaves has a direct connection to optimality of the algorithm, we monitor how the leaves evolve. An optimal algorithm will lead to a dictionary where the type of the reconstructed sequence is close to "optimal reproduction type". Since the underlying distribution in unknown, the optimal reproduction type is also unknown. However, in chapter 6, we develop a method to learn the optimal reproduction type. In a comprehensive set of simulations, we demonstrate different properties of the algorithm. We plot compression rate, running time, type of the leaves in the dictionary, number of the leaves with an specific type, distortion evolving and evolution of the length of the leaves.

Part I

Universal Compression of Unknown Sources

2

Preliminaries and Background

2.1 Introduction

A number of statistical inference problems of significant contemporary interest, such as text classification, language modeling, and DNA microarray analysis are what are called *large alphabet* problems. They require inference on sequences of symbols where the symbols come from a set (*alphabet*) with size comparable or even larger than the sequence length. For instance, language models for speech recognition estimate distributions over English words using text samples much smaller than the English vocabulary.

An abstraction behind several of these problems is universal compression over large alphabets. The general idea here is to model the problem at hand with a collection of models $\mathcal{P}$ instead of a single distribution. The model underlying the data is assumed or known to belong to the collection $\mathcal{P}$, but the exact identity of the model remains unknown. Instead, we aim to use a universal description of data.

2.2 Redundancy

Let $\mathcal{P}$ be a collection of distributions over $\mathbb{N}$ and $\mathcal{P}^n$ be the set of distributions over length-n sequences obtained by *i.i.d.* sampling from $\mathcal{P}$. Note that any finite length sequence or any collection of finite length sequences corresponds to a subset of infinite length sequences. The collection of such subsets corresponding to finite length sequences and collections of finite length sequences forms a semi-algebra [12]. Therefore, for all p, the distributions obtained on finite length sequences by *i.i.d.* sampling can be extended to a measure over infinite length sequences. Let $\mathcal{P}^\infty$ be the collection of all such measures over infinite length sequences of $\mathbb{N}$ obtained by *i.i.d.* sampling from a distribution in $\mathcal{P}$. In a slight abuse of notation to simplify exposition where possible, we use the same symbol p to indicate the distribution in $\mathcal{P}$, $\mathcal{P}^n$, or the measure in $\mathcal{P}^\infty$.

Let q be a measure over infinite sequences of naturals, and define for any $p \in \mathcal{P}^n$, the redundancy [1]

$$R_n(p, q) = \sum_{X^n \in \mathbb{N}^n} p(X^n) \log \frac{p(X^n)}{q(X^n)} \overset{\text{def}}{=} D(p_{X^n} || q_{X^n}), \tag{2.1}$$

where $D()$ above denotes the KL divergence between the length n distributions induced by *i.i.d.* sampling from p to the length n distribution induced by the measure q. Define

$$R_n = R(\mathcal{P}^n) = \inf_q \sup_{p \in \mathcal{P}^n} R_n(p, q).$$

the redundancy of length-n sequences, or *length-n i.i.d. redundancy* or simply length-n

[1] All the logarithms are in base 2, unless otherwise specified.

redundancy. The *single letter redundancy* refers to the special case when $n = 1$.

Our primary goal is to understand the connections between the single letter redundancy on the one hand and the behavior of length-n *i.i.d.* redundancy on the other. Length-n redundancy is the capacity of a channel from $\mathcal{P}$ to $\mathbb{N}^n$, where the conditional probability distribution over $\mathbb{N}^n$ given $p \in \mathcal{P}$ is simply the distribution p over length-n sequences. Roughly speaking, it quantifies how much information about the source we can extract from the sequence.

We will often speak of the *per-symbol* length-n redundancy, which is simply length-n redundancy normalized by n *i.e.,* $R(\mathcal{P}^n)/n$. Furthermore, the limit $\limsup_{n\to\infty} R(\mathcal{P}^n)/n$ is the *asymptotic per-symbol redundancy.* Whether the asymptotic per-symbol redundancy is 0 [2] is in many ways a litmus test for compression, estimation and other related problems. Loosely speaking, if $R(\mathcal{P}^n)/n \to 0$ the redundancy-capacity interpretation [13] mentioned above implies that after a point, there is little further information to be learnt when we see an additional symbol no matter what the underlying source is. In this sense, this is the case where we can actually *learn* the underlying model at a uniform rate over the entire class.

A collection $\mathcal{P}^n$ is *weakly compressible* if there exists a measure q over infinite sequences of naturals such that for all $p \in \mathcal{P}^n$

$$\lim_{n\to\infty} \frac{1}{n} R_n(p, q) = 0.$$

A collection $\mathcal{P}^n$ is *strongly compressible* if there exists a measure q such that

$$\lim_{n\to\infty} \sup_{p\in\mathcal{P}^n} \frac{1}{n} R_n(p, q) = 0.$$

One can consider weak vs strong compressibility as pointwise convergence in contrast to uniform convergence. While weak compressibility needs $R_n(p, q)$to go to zero for each p as $n \to \infty$, strong compressibility needs the uniform convergence of $R_n(p, q)$ toward to zero as

[2] We will equivalently say the asymptotic per-symbol redundancy *diminishes to 0* to keep in line with prior literature.

$n \to \infty$. In fact, we can use Egorov's theorem to connect weak compressibility and strong compressibility using Lemma 1.

Lemma 1. [Egorov's Theorem] Let $\{f_n(\theta)\}, \theta \in \Theta$ be a sequence of measurable functions on measurable space (Θ, Σ, μ) where μ is a finite measure and $f(\theta)$ be a measurable functions on this space. If $\{f_n(\theta)\}$ converges to $f(\theta)$ pointwise, then for every $\epsilon > 0$, there is a subset $B \subset \Theta$ such that $\mu(B) < \epsilon$ and $\{f_n(\theta)\}$ converges to $f(\theta)$ uniformly on $B^c = \Theta - B$. □

2.3 Patterns

Recent work [14] has formalized a similar framework for countably infinite alphabets. This framework is based on the notion of *patterns* of sequences that abstract the identities of symbols, and indicate only the relative order of appearance. For example, the pattern of PATTERN is 1233456. The kth distinct symbol of a string is given an index k when it first appears, and that index is used every time the symbol appears henceforth. The crux of the patterns approach is to consider the set of measures induced over patterns of the sequences instead of considering the set of measures $\mathcal{P}$ over infinite sequences,

Denote the pattern of a string $\mathbf{x}$ by $\Psi(\mathbf{x})$. There is only one possible pattern of strings of length 1 (no matter what the alphabet, the pattern of a length-1 string is 1), two possible patterns of strings of length 2 (11 and 12), and so on. The number of possible patterns of length n is the nth Bell number [14] and we denote the set of all possible length n patterns by Ψ^n. The measures induced on patterns by a corresponding measure p on infinite sequences of natural numbers assigns to any pattern ψ a probability

$$p(\psi) = p(\{\mathbf{x} : \Psi(\mathbf{x}) = \psi\}).$$

In [14] the length-n pattern redundancy,

$$\inf_q \sup_{p \in \mathcal{P}^n} E_p \log \frac{p(\Psi(X^n))}{q(\Psi(X^n))},$$

was shown to be upper bounded by $\pi(\log e)\sqrt{\frac{2n}{3}}$. It was also shown in [27] that there is a measure q over infinite length sequences which satisfies for all n simultaneously

$$\sup_{p\in\mathcal{P}^n}\sup_{X^n}\log\frac{p(\Psi(X^n))}{q(\Psi(X^n))}\leq\pi(\log e)\sqrt{\frac{2n}{3}}+\log(n(n+1)).$$

Let the measure induced on patterns by q be denoted as $q_{_\Psi}$ for convenience.

We can interpret the probability estimator $q_{_\Psi}$ as a sequential prediction procedure that estimates the probability that the symbol X_{n+1} will be "new" (has not appeared in X_1^n), and the probability that X_{n+1} takes a value that has been seen so far. This view of estimation also appears in the statistical literature on Bayesian nonparametrics that focuses on exchangeability. Kingman [15] advocated the use of *exchangeable random partitions* to accommodate the analysis of data from an alphabet that is not bounded or known in advance. A more detailed discussion of the history and philosophy of this problem can be found in the works of Zabell [16, 17] collected in [18].

2.4 Weak Compression Over Infinite Alphabets

Although arbitrary collections of stationary ergodic distributions over finite alphabets are weakly compressible, Kieffer [1] showed that the collection of *i.i.d.* distributions over $\mathbb{N}$ is not even weakly compressible. Indeed the finiteness of single letter redundancy characterizes weak compressibility. Any collection of stationary ergodic measures over infinite sequences is weakly compressible iff $R_1<\infty$.

R_1 being finite, however, is not sufficient for strong compression guarantees to hold even when while dealing with *i.i.d.* sampling. We reproduce the following Example 1 from [19] to illustrate the pitfalls with strong compression, and to motivate the notion of *tail redundancy* that will be central to our main result. Proposition 1 shows that the collection in the example below has finite single letter redundancy, but Proposition 2 shows that its length n

redundancy does not diminish to zero as $n \to \infty$.

Example 1. Partition the set $\mathbb{N}$ into $T_i = \{2^i, \ldots, 2^{i+1} - 1\}$, $i \in \mathbb{N}$. Recall that T_i has 2^i elements. For all $0 < \epsilon \leq 1$, let $n_\epsilon = \lfloor \frac{1}{\epsilon} \rfloor$. Let $1 \leq j \leq 2^{n_\epsilon}$ and let $p_{\epsilon,j}$ be a distribution on $\mathbb{N}$ that assigns probability $1 - \epsilon$ to the number 1 (or equivalently, to the set T_0), and ϵ to the jth smallest element of T_{n_ϵ}, namely the number $2^{n_\epsilon} + j - 1$. $\mathcal{B}$ (mnemonic for binary, since every distribution has at support of size 2) is the collection of distributions $p_{\epsilon,j}$ for all $\epsilon > 0$ and $1 \leq j \leq 2^{n_\epsilon}$. $\mathcal{B}^\infty$ is the set of measures over infinite sequences of numbers corresponding to *i.i.d.* sampling from $\mathcal{B}$. □

We first verify that the single letter redundancy of $\mathcal{B}$ is finite.

Proposition 1. Let q be a distribution that assigns $q(T_i) = \frac{1}{(i+1)(i+2)}$ and for all $j \in T_i$,

$$q(j|T_i) = \frac{1}{|T_i|}.$$

Then

$$\sup_{p \in \mathcal{B}} \sum_{x \in \mathbb{N}} p(x) \log \frac{p(x)}{q(x)} \leq 2. \qquad \square$$

However, the redundancy of compressing length-n sequences from $\mathcal{B}^\infty$ scales linearly with n.

Proposition 2. For all $n \in \mathbb{N}$,

$$\inf_q \sup_{p \in B^\infty} E_p \log \frac{p(X^n)}{q(X^n)} \geq n \left(1 - \frac{1}{n}\right)^n.$$

Proof Let the set $\{1^n\}$ denote a set containing a length-n sequence of only ones. For all n, define 2^n pairwise disjoint sets S_i of $\mathbb{N}^n$, $1 \leq i \leq 2^n$, where

$$S_i = \{1, 2^n + i - 1\}^n - \{1^n\}$$

is the set of all length-n strings containing at most two numbers (1 and $2^n + i - 1$) and at least one occurrence of $2^n + i - 1$. Clearly, for distinct i and j between 1 and 2^n, S_i and S_j

are disjoint. Furthermore, the measure $p_{\frac{1}{n},i} \in \mathcal{B}^\infty$ assigns S_i the probability

$$p_{\frac{1}{n},i}(S_i) = 1 - \left(1 - \frac{1}{n}\right)^n > 1 - \frac{1}{e}.$$

From Lemma 3 in [19], it follows that length-n redundancy of $\mathcal{B}^\infty$ is lower bounded by

$$\left(1 - \frac{1}{e}\right) \log 2^n = n\left(1 - \frac{1}{e}\right). \qquad \square$$

2.5 Worst-Case redundancy

It is possible to define an even more stringent notion—a *worst-case-regret.* For length-n sequences, worst case regret is defined as

$$\inf_q \sup_{p \in \mathcal{P}^n} \sup_{X^n} \log \frac{p(X^n)}{q(X^n)},$$

single letter regret is the special case where $n = 1$ and asymptotic per-symbol regret is the limit as $n \to \infty$ of the length-n regret normalized by n.

Recent work in [20] has shown that if the single letter worst-case redundancy of $\mathcal{P}$ is finite, then length-n *i.i.d.* sequences from $\mathcal{P}$ can be compressed with worst-case redundancy that is sublinear in n. Since the average case redundancy of any scheme is upper bounded by its worst case redundancy, it follows then that there is a universal measure q over infinite length strings of natural numbers such that

$$\lim_{n\to\infty} \sup_{p \in \mathcal{P}} \frac{1}{n} E_p \log \frac{p(X^n)}{q(X^n)} \to 0. \tag{2.2}$$

However, what happens when the worst case redundancy of a class $\mathcal{P}$ is not finite? Since the underlying support of distributions in $\mathcal{P}$ is countably infinite, it is easy to construct collections $\mathcal{P}$ whose single letter average-case redundancy is finite, *i.e.,*

$$\inf_q \sup_{p\in\mathcal{P}} \sum_{x\in\mathbb{N}} p(x)\log\frac{p(x)}{q(x)} < \infty \tag{2.3}$$

even when the single letter worst-case redundancy is not. Suppose, we only knew that the single letter average-case redundancy of $\mathcal{P}$ is finite, namely that (2.3) holds. We note that in general, the guarantee (2.2) need not hold (see Example 1).

In finite alphabet regime length$-n$ redundancy and worst case redundancy have a same behavior. They grow in a same order with length of the sequence. However, for infinite alphabet it is possible to construct classes with finite length$-n$ redundancy and infinite regret. In the next example we construct a collection of distributions where the worst case regret is infinite but average case redundancy is finite. Other similar examples can be found in [21] and [22].

Example 2. Let $p_k(x)$, $x\in\mathcal{X}, \mathcal{X}=\{2,3,4,\ldots\}$,

$$p_k(x)=\begin{cases} 1-\frac{1}{\log k} & x=2,\\ \frac{1}{k\log k} & k\le x<2k,\\ 0 & \text{other wise.}\end{cases} \qquad \square$$

and $\mathcal{P}=\{p_2,p_3,\ldots\}$. Then the worst-case redundancy is

$$\log\sum_{x\in\mathcal{X}}\sup_{p_k\in\mathcal{P}} p_k(x) \ge \log\sum_k \frac{1}{k\log k},$$

and since $\sum_k \frac{1}{k\log k}$ diverges, the worst case redundancy is infinite. To see finiteness of average case redundancy, let $q(x) = \frac{1/a}{x \log^2(x)}$ where $a = \sum_x \frac{1}{x \log^2(x)}$. Then

$$
\begin{aligned}
D(p_k||q) &= (1 - \frac{1}{\log k}) \log 2a(1 - \frac{1}{\log k}) \\
&\quad + \frac{1}{k \log k} \log \frac{\frac{1}{k \log k}}{\frac{1}{ak \log^2 k}} + \frac{1}{k \log k} \log \frac{\frac{1}{k \log k}}{\frac{1}{a(k+1) \log^2 (k+1)}} + \cdots + \frac{1}{k \log k} \log \frac{\frac{1}{k \log k}}{\frac{1}{a2k \log^2 2k}} \\
&\le (1 - \frac{1}{\log k}) \log 2a(1 - \frac{1}{\log k}) + \frac{1}{k \log k} \log \frac{\frac{1}{k \log k}}{\frac{1}{a2k \log^2 k^2}} + \cdots + \frac{1}{k \log k} \log \frac{\frac{1}{k \log k}}{\frac{1}{a2k \log^2 k^2}} \\
&= (1 - \frac{1}{\log k}) \log 2a(1 - \frac{1}{\log k}) + \frac{\log 8a \log k}{\log k} \\
&\le 1 + \log 16a^2.
\end{aligned}
$$

□

2.6 Bayes Redundancy

A well known lower bound on the average case redundancy relates it to Bayes redundancy of any given prior. This result can be obtained from general minimax theorem. Here we provide a version from [23].

Lemma 2. Let $\mathcal{M}(x^n)$ show the set of all probability measure on x^n. Denotes elements of $\mathcal{P}^n$ as $p_\theta, \theta \in \Theta$. Then the redundancy is lower bounded by Bayes redundancy of any given prior π on Θ. i.e.

$$R(\mathcal{P}^n) \ge \inf_{q \in \mathcal{M}(x^n)} E_\pi D(p_\theta||q)$$

□

Lemma 3. [Minimax Theorem [23]] Let $\mathcal{M}(x^n)$ show the set of all probability measure on X^n. Denotes elements of $\mathcal{P}^n$ as $p_{\theta \in \Theta}$. Then

$$R(\mathcal{P}^n) = \sup_{\pi \in \mathcal{M}(\Theta)} \inf_{q \in \mathcal{M}(x^n)} E_\pi D(p_\theta||q) = \sup_{p_\theta \in \mathcal{P}^n} \inf_{q \in \mathcal{M}(x^n)} D(p_\theta||q)$$

□

2.7 Metric Entropy

A prior work on connecting single letter metric to length$-n$ redundancy is in [21], where authors find lower and upper bound on redundancy using Hellinger distance.

Definition 1. [Hellinger Distance] Let p_1 and p_2 be two distributions in $\mathcal{P}$. The Hellinger distance h is defined as

$$h(p_1, p_2) = \sum_{x \in \mathbb{N}} \left(\sqrt{p_1(x)} - \sqrt{p_2(x)} \right)^2 .$$

□

Lemma 4. [24], [21] Let π be any prior on $\mathcal{P}^n$ then

$$R(\mathcal{P}^n) \geq E_{p_1 \sim \pi} \left[- \log E_{p_2 \sim \pi} e^{-n \frac{h(p_1, p_2)}{2}} \right],$$

where p_1 and p_2 drawn independently according to π.

Proof See [21]. □

2.7.1 Totally Boundedness

To study the connection of length$-n$ redundancy to single letter redundancy, Haussler and Opper [21] characterize collections with finite single letter redundancy but infinite length$-n$ redundancy using totally boundedness of a collection.

Definition 2. [Totally Bounded Set [21]] Let (S, ρ) be any complete separable metric space. A partition Π of set S is a collection of disjoint Borel subsets of S such that their union is S. Then diameter of a subset $A \subset S$ is $d(A) = \sup_{x,y \in A} \rho(x, y)$ and diameter of partition Π is supremum of diameters of the sets in the partition. For $\epsilon > 0$, let $\mathcal{D}_\epsilon(S, \rho)$ be the cardinality of the smallest finite partition of S of diameter at most ϵ. We say S is totally bounded if $\mathcal{D}_\epsilon(S, \rho) < \infty$ for all $\epsilon > 0$. □

Note that if we use Hellinger distance h as a metric, $(\mathcal{P}, h)$ is a metric space.

Lemma 5. [21] If length$-n$ redundancy is finite it can grow at most linearly in n. If $(\mathcal{P}, h)$ is not *totally bounded* and single letter redundancy is finite then $\liminf_{n\to\infty} \frac{1}{n} R(\mathcal{P}^n)$ is bounded away from zero and $\limsup_{n\to\infty} \frac{1}{n} R(\mathcal{P}^n) < \infty$.

Proof See [21, Theorem 4, part 5]. □

In the next example we construct collection $\mathcal{U}$ that is totally bound.

Construction Let $\mathcal{U}$ be a countable collection of distributions p_k, $k \geq 1$ over $\mathbb{Z}^+ = \mathbb{N} \cup \{0\}$, where

$$p_k(x) = \begin{cases} 1 - \frac{1}{k^2} & x = 0, \\ \frac{1}{k^2 2^{k^2}} & 1 \leq x \leq 2^{k^2}. \end{cases}$$ □

Proposition 3. $(\mathcal{U}, h)$ is totally bounded.

Proof To show that $(\mathcal{U}, h)$ is totally bounded we need to show that for any $\epsilon > 0$ there exists a partition on $\mathcal{U}$ with diameter ϵ and finite cardinality. For any given $\epsilon > 0$ let $m = \sqrt{\frac{3}{\epsilon}} + 1$. We construct Partition Π so that it packs p_1 through p_m in m singletons and all other distributions in the collection in a single set. Therefore the cardinality of the partition is $m + 1 < \infty$. Now we show that the diameter of each set in Π is less than ϵ.

For singleton the diameter is zero. For a single set containing all distributions in the collection we can bound the diameter as below. Let $i < j$ then

$$\begin{aligned} h(p_i, p_j) &= \sum_{x \in \mathbb{Z}^+} \left(\sqrt{p_i(x)} - \sqrt{p_j(x)} \right)^2 \\ &= \left(\sqrt{1 - \frac{1}{i^2}} - \sqrt{1 - \frac{1}{j^2}} \right)^2 + 2^{i^2} \left(\sqrt{\frac{1}{i^2 2^{i^2}}} - \sqrt{\frac{1}{j^2 2^{j^2}}} \right)^2 + (2^{j^2} - 2^{i^2}) \frac{1}{j^2 2^{j^2}} \\ &\leq \frac{1}{i^2} + \frac{1}{i^2} + \frac{1}{j^2} \\ &\leq \frac{3}{m^2} < \epsilon. \end{aligned}$$

Where the last step is using the fact that $i > m$.

□

3

Length-n Redundancy

3.1 Tightness

We focus on the single letter redundancy in this section, and explore the connections between the single letter redundancy of a collection $\mathcal{P}$ and the tightness of $\mathcal{P}$.

Lemma 6. A collection $\mathcal{P}$ with bounded length-n redundancy is tight. Namely, if the single letter redundancy of $\mathcal{P}$ is finite, then for any $\gamma > 0$

$$\sup_{p \in \mathcal{P}} F_p^{-1}(1-\gamma) < \infty.$$

Proof $\mathcal{P}$ has bounded single letter redundancy. Let q be a distribution over $\mathbb{N}$ such that

$$\sup_{p\in\mathcal{P}} D(p||q) < \infty,$$

and we define $R = \sup_{p\in\mathcal{P}} D(p||q)$ where $D(p||q)$ is Kullback-Leibler distance between p and q. It follows that for all $p \in \mathcal{P}$ and any m,

$$p(\left|\log\frac{p(X)}{q(X)}\right| > m) \le (R + (2\log e)/e)/m,$$

To see the above, note that if S is the set of all numbers such that $p(x) < q(x)$, a well-known convexity argument shows that

$$\sum_x p(x)\log\frac{p(x)}{q(x)} \ge p(S)\log\frac{p(S)}{q(S)} \ge -\frac{\log e}{e}.$$

We prove the lemma by contradiction. Pick m so large that $(R + (2\log e)/e)/m < \gamma/2$. For all p, we show that

$$p\big(x : x \ge F_q^{-1}(1-\gamma/2^{m+1})\big) \le \gamma.$$

To see the above, observe that we can split the tail $x \ge F_q^{-1}(1-\gamma/2^{m+1})$ into two parts (i) numbers x such that $\log\frac{p(x)}{q(x)} > m$. This set has probability $< \gamma/2$ under p. (ii) remaining numbers x such that $\log\frac{p(x)}{q(x)} < m$. This set has probability $\le \gamma/2^{m+1}$ under q, and therefore probability $\le \gamma/2$ under p. The lemma follows. □

The converse is not necessarily true. Tight collections need not have finite single letter redundancy as the following example demonstrates.

Construction Consider the following collection $\mathcal{I}$ of distributions over $\mathbb{N}$. First partition the set of natural numbers into the sets T_i, $i \in \mathbb{N}$, where

$$T_i = \{2^i, \ldots, 2^{i+1} - 1\}.$$

Note that $|T_i| = 2^i$. Now, $\mathcal{I}$ is the collection of all possible distributions that can be formed as follows. For all $i \geq 1$, we pick exactly one element of T_i and assign it probability $1/(i(i+1))$. Note that the set $\mathcal{I}$ is uncountably infinite. □

Corollary 7. The set $\mathcal{I}$ of distributions is tight.

Proof For all $p \in \mathcal{I}$,

$$\sum_{\substack{x \geq 2^k \\ x \in \mathbb{N}}} p(x) = \frac{1}{k+1},$$

namely, all tails are uniformly bounded over the collection $\mathcal{I}$. Put another way, for all $\delta > 0$ and all distributions $p \in \mathcal{I}$,

$$F_p^{-1}(1-\delta) \leq 2^{\lfloor \frac{1}{\delta} \rfloor} - 1.$$

□

On the other hand,

Proposition 4. The collection $\mathcal{I}$ does not have finite redundancy.

Proof Suppose q is any distribution over $\mathbb{N}$. We will show that $\exists p \in \mathcal{I}$ such that

$$\sum_{\substack{x \geq 1 \\ x \in \mathbb{N}}} p(x) \log \frac{p(x)}{q(x)}$$

is not finite. Since the entropy of every $p \in \mathcal{I}$ is finite, we just have to show that for any distribution q over $\mathbb{N}$, there $\exists p \in \mathcal{I}$ such that

$$\sum_{\substack{x \geq 1 \\ x \in \mathbb{N}}} p(x) \log \frac{1}{q(x)}$$

is not finite.

Consider any distribution q over $\mathbb{N}$. Observe that for all i, $|T_i| = 2^i$. It follows that for all i there is $x_i \in T_i$ such that

$$q(x_i) \leq \frac{1}{2^i}.$$

But by construction, $\mathcal{I}$ contains a distribution p that has for its support $\{x_i : i \geq 1\}$ identified above. Furthermore p assigns

$$p(x_i) = \frac{1}{i(i+1)} \qquad \forall\, i \geq 1.$$

The KL divergence from p to q is not finite and the Lemma follows. □

The collection of monotone distributions with finite entropy is known to be weakly compressible. We now use Lemma 6 to verify that it is not strongly compressible.

Example 3. Let $\mathcal{M}$ be the collection of monotone distributions over $\mathbb{N}$ with finite entropy. Let $\mathcal{M}^\infty$ be the set of all *i.i.d.* processes obtained from distributions in $\mathcal{M}$. For all $p \in \mathcal{M}$ and all numbers n, we have

$$p(n) \leq \frac{1}{n}.$$

So

$$\sum_{n\geq 1} p(n) \log n \leq \sum_{n\geq 1} p(n) \log \frac{1}{p(n)} \leq \infty,$$

and from Kieffer's condition $\mathcal{M}^\infty$ is weakly compressible.

However, it is easy to verify that $\mathcal{M}$ is not tight. To see this, consider the collection $\mathcal{U}$ of all uniform distributions over finite supports of form $\{m, m+1, \ldots, M\}$ for all positive integers m and M with $m \leq M$. Let $\mathcal{U}^\infty$ be the set of all *i.i.d.* processes with one dimensional marginal from $\mathcal{U}$. Consider distributions of form $p' = (1-\epsilon)p + \epsilon q$ where $q \in \mathcal{U} \cap \mathcal{M}$ is a monotone uniform distribution and $\epsilon > 0$. The ℓ_1 distance between p and q is $\leq 2\epsilon$. For all $M > 0$ and $\delta \leq \epsilon$, we can pick $q \in \mathcal{U}$ over a sufficiently large support such that $F_{p'}^{-1}(1-\delta) > M$, so $\mathcal{M}$ is not tight.

Since $\mathcal{M}$ is not tight, from Lemma 6 its single letter redundancy is not finite. Hence the length-n redundancy cannot be finite for any n and $\mathcal{M}^\infty$ is not strongly compressible. □

3.2 Sufficient Condition

We study how the single letter properties of a collection $\mathcal{P}$ of distributions influences the compression of length-n strings obtained by *i.i.d.* sampling from distributions in $\mathcal{P}$. Namely, we try to characterize when the length-n redundancy of $\mathcal{P}^\infty$ grows sublinearly in the block-length n.

Lemma 8. Let $\mathcal{P}$ be a collection of distributions over a countable support $\mathcal{X}$. For some $m \ge 1$, consider m pairwise disjoint subsets $S_i \subset \mathcal{X}$ $(1 \le i \le m)$ and let $\delta > 1/2$. If there exist $p_1, \ldots, p_m \in \mathcal{P}$ such that

$$p_i(S_i) \ge \delta,$$

then for all distributions q over $\mathcal{X}$,

$$\sup_{p \in \mathcal{P}} D(p||q) \ge \delta \log m.$$

In particular if there are an infinite number of sets S_i, $i \ge 1$ and distributions $p_i \in \mathcal{P}$ such that $p_i(S_i) \ge \delta$, then the redundancy is infinite.

Proof This is a simplified formulation of the *distinguishability* concept in [13]. For a proof, see *e.g.* [25]. □

W show a sufficient condition on single letter marginals of $\mathcal{P}$ and its redundancy that allows for *i.i.d.* length-n redundancy of $\mathcal{P}^\infty$ to grow sublinearly with n. This condition is, however, not necessary—and the characterization of a condition that is both necessary and sufficient is as yet open.

For all $\epsilon > 0$, let $A_{p,\epsilon}$ be the set of all elements in the support of p with probability $\ge \epsilon$, and let $T_{p,\epsilon} = \mathbb{N} - A_{p,\epsilon}$. Let $G_0 = \{\phi\}$ where ϕ denotes the empty string. For all i, the sets

$$G_i = \{x^i : A_{p, \frac{2\ln(i+1)}{i}} \subseteq \{x_1, x_2, \ldots, x_i\}\}$$

where in a minor abuse of notation, we use $\{x_1, \ldots, x_i\}$ to denote the set of distinct symbols in the string x_1^i. Let $B_0 = \{\}$ and let $B_i = \mathbb{N}^i - G_i$. Observe from an argument similar to the coupon collector problem that

Lemma 9. For all $i \geq 2$,

$$p(B_i) \leq \frac{i+1}{2\ln(i+1)}\left(1 - \frac{2\ln(i+1)}{i}\right)^i \leq \frac{1}{(i+1)\ln(i+1)}. \qquad \square$$

Theorem 10. Suppose $\mathcal{P}$ is a collection of distributions over $\mathbb{N}$. Let the entropy of $p \in \mathcal{P}$, be uniformly bounded over the entire collection, and in addition let the redundancy of the collection be finite. Namely,

$$\sup_{p\in\mathcal{P}} \sum_{x\in\mathbb{N}} p(x)\log\frac{1}{p(x)} \overset{\text{def}}{=} H < \infty \quad \text{ and } \exists q_1 \text{ over } \mathbb{N} \text{ s.t.} \quad \sup_{p\in\mathcal{P}} \sum_{x\in\mathbb{N}} p(x)\log\frac{p(x)}{q_1(x)} \overset{\text{def}}{=} R < \infty.$$

Recall that for any distribution p, the set $T_{p,\delta}$ denotes the support of p all of whose probabilities are $< \delta$. Let

$$\lim_{\delta\to 0}\sup_{p\in\mathcal{P}} \sum_{x\in T_{p,\delta}} p(x)\log\frac{1}{p(x)} = 0 \quad \text{ and } \exists q_1 \text{ over } \mathbb{N} \text{ s.t.} \quad \lim_{\delta\to 0}\sup_{p\in\mathcal{P}} \sum_{x\in T_{p,\delta}} p(x)\log\frac{p(x)}{q_1(x)} = 0. \tag{3.1}$$

Then, the redundancy of length-n distributions obtained by *i.i.d.* sampling from distributions in $\mathcal{P}$, denoted by $R_n(\mathcal{P}^\infty)$, grows sublinearly

$$\limsup_{n\to\infty} \frac{1}{n} R_n(\mathcal{P}^\infty) = 0.$$

Proof Let q_Ψ be the optimal universal pattern encoder over patterns of *i.i.d.* sequences from Section 2.3. Recall that the redundancy of $\mathcal{P}$ is finite, and that q_1 is the universal distribution over $\mathbb{N}$ that attains redundancy R for $\mathcal{P}$.

In what follows x^i represents a string $x_1, \ldots, x_i$, and x^0 denotes the empty string. For all n, we denote $\Psi(x^n) = \psi_1, \ldots, \psi_n$ and $\Psi(X^n) = \Psi_1, \ldots, \Psi_n$.

We consider a universal encoder as follows:

$$\begin{aligned}
q(x^n) &= q(x^n, \Psi(x^n)) \\
&= q(\psi_1, x_1, \psi_2, x_2, \ldots, \psi_n, x_n) \\
&= \prod_{i\geq 1} q(\psi_i|\psi_1^{i-1}, x_1^{i-1}) \prod_{j\geq 1} q(x_j|\psi_1^j, x_1^{j-1}) \\
&\overset{\text{def}}{=} \prod_{i\geq 1} q_\Psi(\psi_i|\psi_1^{i-1}) \prod_{j\geq 1} q(x_j|\psi_1^j, x_1^{j-1}).
\end{aligned}$$

Furthermore we define for all $x_1^{i-1} \in \mathbb{N}^{i-1}$ and all $\psi^i \in \Psi^i$ such that $\psi^{i-1} = \Psi(x^{i-1})$,

$$q(x_i|\psi_1^i, x_1^{i-1}) \overset{\text{def}}{=} \begin{cases} 1 & \text{if } x_i \in \{x_1, \ldots, x_{i-1}\} \text{ and } \Psi(x^i) = \psi^i \\ q_1(x_i) & \text{if } x_i \notin \{x_1, \ldots, x_{i-1}\} \text{ and } \Psi(x^i) = \psi^i. \end{cases}$$

Namely, we use an optimal universal pattern encoder over patterns of *i.i.d.* sequences, and encode any new symbol using a universal distribution over $\mathcal{P}$. We now bound the redundancy of q as defined above. We have for all $p \in \mathcal{P}^\infty$,

$$\begin{aligned}
E_p \log \frac{p(X^n)}{q(X^n)} &= \sum_{x^n} p(x^n) \log \prod_{i\geq 1} \frac{p(\psi_i|\psi_1^{i-1}, x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})} \prod_{j\geq 1} \frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})} \\
&= \sum_{x^n} p(x^n) \sum_{i=1}^n \log \frac{p(\psi_i|\psi_1^{i-1}, x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})} + \sum_{x^n} p(x^n) \sum_{j=1}^n \log \frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})}.
\end{aligned}$$

Since ψ_1 is always 1, $p(\psi_1) = q_\Psi(\psi_1) = 1$. Therefore, we have

$$\sum_{x^n} p(x^n) \sum_{i=1}^n \log \frac{p(\psi_i|\psi_1^{i-1}, x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})} = \sum_{x^n} p(x^n) \sum_{i=2}^n \log \frac{p(\psi_i|\psi_1^{i-1}, x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})}.$$

The first term, normalized by n, can be upper bounded by as follows

$$\begin{aligned}\frac{1}{n}\sum_{x^n} p(x^n)\sum_{i=2}^{n}\log\frac{p(\psi_i|\psi_1^{i-1},x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})}&\\ &\leq \frac{1}{n}\sum_{i=2}^{n}\sum_{x_1^i} p(x_1^i)\log\frac{p(\psi_i|\psi_1^{i-1},x_1^{i-1})}{p(\psi_i|\psi_1^{i-1})}+\frac{1}{n}\left(\pi\log e\sqrt{\frac{2n}{3}}+\log n(n+1)\right)\\ &=\frac{1}{n}\sum_{i=2}^{n}(H(\Psi_i|\Psi_1^{i-1})-H(\Psi_i|X_1^{i-1}))+\frac{1}{n}\left(\pi\log e\sqrt{\frac{2n}{3}}+\log n(n+1)\right)\\ &\leq \frac{1}{n}(nH)-\frac{1}{n}\sum_{i=2}^{n}H(\Psi_i|X_1^{i-1}))+\frac{1}{n}\left(\pi\log e\sqrt{\frac{2n}{3}}+\log n(n+1)\right)\end{aligned}$$

where the last inequality follows since

$$H(\Psi^n)\leq H(X^n)=nH.$$

Now for $i\geq 2$,

$$\begin{aligned}H-H(\Psi_i|X_1^{i-1})&=\sum_{x^{i-1}}p(x_1^{i-1})\sum_{x\notin\{x_1,\ldots,x_{i-1}\}}p(x)\log\frac{1}{p(x)}\\ &\leq p(G_{i-1})\sum_{x\in T_{p,2\frac{\ln i}{i-1}}}p(x)\log\frac{1}{p(x)}+p(B_{i-1})H\\ &\leq \sum_{x\in T_{p,2\frac{\ln i}{i-1}}}p(x)\log\frac{1}{p(x)}+\frac{H}{i\ln i}.\end{aligned}$$

We have split the length $i-1$ sequences into the sets G_{i-1} and B_{i-1} and use separate bounds on each set that hold uniformly over the entire model collection. The last inequality above follows from Lemma 9. From condition (3.1) of the Theorem, we have that

$$\lim_{i\to\infty}\sup_{p\in\mathcal{P}}\sum_{x\in T_{p,2\frac{\ln i}{i-1}}}p(x)\log\frac{1}{p(x)}=0.$$

Therefore we have

$$\lim_{n\to\infty} \sup_{p\in\mathcal{P}} \frac{1}{n} \sum_{i=2}^{n} \left(\sum_{x\in T_{p,2\frac{\ln i}{i-1}}} p(x)\log\frac{1}{p(x)} + \frac{H}{i\ln i} \right) \le \lim_{n\to\infty} \frac{1}{n} \sum_{i=2}^{n} \left(\sup_{p\in\mathcal{P}} \sum_{x\in T_{p,2\frac{\ln i}{i-1}}} p(x)\log\frac{1}{p(x)} + \frac{H}{i\ln i} \right)$$
$$\overset{(a)}{=} 0.$$

The first term on the left in the first equation above is non-negative, hence the limit above has to equal 0. The equality (a) follows from Cesaro's lemma asserting that for any sequence $\{a_i, i \ge 1\}$ with $a_i < \infty$ for all i, if $\lim_{i\to\infty} a_i$ exists then

$$\lim_{i\to\infty} a_i = \lim_{n\to\infty} \frac{1}{n}\sum_{j=1}^{n} a_j.$$

Therefore,

$$\lim_{n\to\infty} \sup_{p\in\mathcal{P}} \frac{1}{n} \sum_{x^n} p(x^n) \sum_{i=2}^{n} \log\frac{p(\psi_i|\psi_1^{i-1}, x_1^{i-1})}{q_\Psi(\psi_i|\psi_1^{i-1})} = 0.$$

For the second term, observe that

$$\sum_{x^n} p(x^n) \sum_{j=1}^{n} \log\frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})} = R + \sum_{x^n} p(x^n) \sum_{j=2}^{n} \log\frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})}.$$

Furthermore,

$$\begin{aligned}
\sum_{x^n} p(x^n) \sum_{j=2}^{n} \log\frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})} &= \sum_{j=2}^{n} \sum_{x^j} p(x^j) \log\frac{p(x_j|\psi_1^j, x_1^{j-1})}{q(x_j|\psi_1^j, x_1^{j-1})} \\
&\le \sum_{j=2}^{n} \sum_{x^{j-1}} p(x^{j-1}) \sum_{x\notin\{x_1,\ldots,x_{i-1}\}} p(x_j)\log\frac{p(x_j)}{q_1(x_j)} \\
&\le \sum_{j=2}^{n} \left(p(G_{j-1}) \sum_{x_j\in T_{p,\frac{2\ln j}{j-1}}} p(x_j)\log\frac{p(x_j)}{q_1(x_j)} + Rp(B_{j-1}) \right) \\
&\le \sum_{j=2}^{n} \sum_{x_j\in T_{p,\frac{2\ln j}{j-1}}} p(x_j)\log\frac{p(x_j)}{q_1(x_j)} + \frac{R}{j\ln j}.
\end{aligned}$$

As before, the last inequality is from Lemma 9. Again from condition (3.1), we have

$$\lim_{j\to\infty}\left(\sup_{p\in\mathcal{P}}\sum_{x_j\in T_{p,\frac{2\ln j}{j-1}}} p(x_j)\log\frac{p(x_j)}{q_1(x_j)}+\frac{R}{j\ln j}\right)=0.$$

Therefore as before

$$\lim_{n\to\infty}\sup_{p\in\mathcal{P}}\frac{1}{n}\left(\sum_{j=1}^{n}\sum_{x_j\in T_{p,\frac{2\ln j}{j-1}}} p(x_j)\log\frac{p(x_j)}{q_1(x_j)}+\sum_{j=2}^{n}\frac{R}{j\ln j}\right)=0$$

as well. The theorem follows. □

A few comments about (3.1) in Theorem 10 are in order. Neither condition automatically implies the other. The set $\mathcal{B}$ of distributions in Example 1 is an example where every distribution has finite entropy, the redundancy of $\mathcal{B}$ is finite,

$$\lim_{\delta\to 0}\sup_{p\in\mathcal{B}}\sum_{x\in T_{p,\delta}} p(x)\log\frac{1}{p(x)}=0 \quad \text{but } \forall q \text{ over } \mathbb{N} \text{ s.t.} \quad \lim_{\delta\to 0}\sup_{p\in\mathcal{B}}\sum_{x\in T_{p,\delta}} p(x)\log\frac{p(x)}{q(x)}>0.$$

We will now construct another set $\mathcal{U}$ of distributions over $\mathbb{N}$ such that every distribution in $\mathcal{U}$ has finite entropy, the redundancy of $\mathcal{U}$ is finite,

$$\lim_{\delta\to 0}\sup_{p\in\mathcal{U}}\sum_{x\in T_{p,\delta}} p(x)\log\frac{1}{p(x)}>0 \quad \text{but } \exists q \text{ over } \mathbb{N} \text{ s.t.} \quad \lim_{\delta\to 0}\sup_{p\in\mathcal{U}}\sum_{x\in T_{p,\delta}} p(x)\log\frac{p(x)}{q(x)}=0. \tag{3.2}$$

At the same time, the length-n redundancy of $\mathcal{U}^\infty$ diminishes sublinearly. This is therefore also an example to show that the conditions in Theorem 10 are only sufficient, but in fact not necessary. It is yet open to find a condition on single letter marginals that is both necessary and sufficient for the asymptotic per-symbol redundancy to diminish to 0.

Example 4.

Construction $\mathcal{U}$ is a countable collection of distributions p_k, $k \geq 1$ where

$$p_k(x) = \begin{cases} 1 - \frac{1}{k^2} & x = 0, \\ \frac{1}{k^2 2^{k^2}} & 1 \leq x \leq 2^{k^2}. \end{cases} \qquad \square$$

The entropy of $p_k \in \mathcal{U}$ is therefore $1 + h\left(\frac{1}{k^2}\right)$. Note that the redundancy of $\mathcal{U}$ is finite too. To see this, first note that

$$\sum_{x \geq 1} \sup_{k \geq 1} p_k(x) \leq \sum_{x \geq 1} \sum_{p_k : k \geq 1} p_k(x) = \sum_{p_k : k \geq 1} \sum_{x \geq 1} p_k(x) = \sum_{p_k : k \geq 1} \frac{1}{k^2} = \frac{\pi^2}{6}. \tag{3.3}$$

Now letting $R \overset{\text{def}}{=} \log\left(\sum_{x \geq 1} \sup_{k \geq 1} p_k(x)\right)$, observe that the distribution

$$q(x) = \begin{cases} 1/2 & x = 0, \\ \frac{\sup_{k \geq 1} p_k(x)}{2^{R+1}} & x \geq 1. \end{cases}$$

satisfies for all $p_k \in \mathcal{U}$

$$\sum_{x \geq 0} p_k(x) \log \frac{p_k(x)}{q(x)} \leq 1 + \frac{R+1}{k^2} \leq R + 2,$$

implying that the redundancy is $\leq R + 2$. Furthermore, (3.3) implies from the results on worst-case regret in [20] that the length-n redundancy of $\mathcal{U}^\infty$ diminishes sublinearly. Now pick an integer $m \geq 1$. We have for all $p \in \mathcal{U}$,

$$\sum_{x \in T_{p, \frac{1}{m^2 2^{m^2}}}} p(x) \log \frac{p(x)}{q(x)} \leq \frac{R+1}{m^2},$$

yet for all $k \geq m$, we have

$$\sum_{x \in T_{p, \frac{1}{m^2 2^{m^2}}}} p_k(x) \log \frac{1}{p_k(x)} = 1.$$

Thus it is easy to see that $\mathcal{U}$ indeed satisfies (3.2). $\square$

4

Tail Redundancy and Its Characterization of Compression of Memoryless Sources

In this chapter, we prove that universal compression of length-n *i.i.d.* sequences from $\mathcal{P}$ is characterized by how well the tails of distributions in $\mathcal{P}$ can be universally described, and we formalize the later as the *tail* redundancy of $\mathcal{P}$. We study the tail redundancy of a collection $\mathcal{P}$ of distributions ans show that the per-symbol redundancy goes to tail redundancy as $n \to \infty$. Therefore, we obtain a single-letter characterization that is both necessary and sufficient for sequences generated *i.i.d.* from a collection $\mathcal{P}$ of distributions over a countably infinite alphabet to be (average-case) strongly compressible. Contrary to the worst case formulation of universal compression, finite single letter (average case) redundancy of $\mathcal{P}$

does not automatically imply that the expected redundancy of describing length-n strings sampled *i.i.d.* from $\mathcal{P}$ grows sub-linearly with n.

4.1 Tail Redundancy

We will develop a series of tools that will help us better understand how the per-symbol redundancy behaves in a wide range of large alphabet cases. In particular, for *i.i.d.* sources, we completely characterize the asymptotic per-symbol redundancy in terms of single letter marginals. Fundamental to our analysis is the understanding of how much complexity lurks in the tails of distributions.

To this end, we define what we call the *tail redundancy.* We assert the basic definition below, but simplify several nuances around it in Section 4.1.1, eventually settling on a operationally workable characterization.

Definition 3. For a collection $\mathcal{P}$ of distributions, define for all $m \geq 1$

$$\mathcal{T}_m(\mathcal{P}) \stackrel{\text{def}}{=} \inf_q \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x) \log \frac{p(x)}{q(x)},$$

where the infimum is over all distributions q over $\mathbb{N}$. We define the *tail redundancy* is defined as

$$\mathcal{T}(\mathcal{P}) \stackrel{\text{def}}{=} \limsup_{m\to\infty} \mathcal{T}_m(\mathcal{P}).$$

The above quantity, $\mathcal{T}_m(\mathcal{P})$ can be negative, and is not a true redundancy as is conventionally understood. However,

$$\tilde{\mathcal{T}}_m(\mathcal{P}) \stackrel{\text{def}}{=} \inf_q \sup_{p\in\mathcal{P}} \left(\sum_{x\geq m} p(x) \log \frac{p(x)}{q(x)} + p(x \geq m) \log \frac{1}{p(x \geq m)} \right)$$

is always non-negative, and can be phrased in terms of a conventional redundancy. To see this, let p' be the distribution over numbers $x \geq m$ obtained from p as $p'(x) = p(x)/p(x \geq m)$, and note that

$$\tilde{\mathcal{T}}_m(\mathcal{P}) = \inf_q \sup_{p\in\mathcal{P}} p(S_m) D\left(p'(x)||q(x)\right). \qquad \square$$

4.1.1 Operational Characterization of Tail Redundancy

We refine the above definitions in several ways. First we prove that

$$\mathcal{T}(\mathcal{P}) = \lim_{m\to\infty} \mathcal{T}_m(\mathcal{P}) = \lim_{m\to\infty} \inf_{q_m} \sup_{p\in\mathcal{P}} \sum_{x>m} p(x)\log\frac{p(x)}{q_m(x)}. \tag{4.1}$$

Next, we show that the limit and inf above can be interchanged, and in addition, that a minimizer exists—namely there is always a distribution over $\mathbb{N}$ that achieves the tail redundancy. This will let us operationally characterize the notions in the definitions above.

Lemma 11. $\tilde{\mathcal{T}}_m(\mathcal{P})$ is non-increasing in m.

Proof Let q be any distribution over $\mathbb{N}$ and $S_m = \{x \geq m\}$. We show that

$$\sup_{p\in\mathcal{P}} \left(\sum_{x\geq m} p(x)\log\frac{p(x)}{q(x)} + p(S_m)\log\frac{1}{p(S_m)} \right) \geq \tilde{\mathcal{T}}_{m+1}(\mathcal{P}),$$

thus proving the lemma.

To proceed, note that without loss of generality we can assume $\sum_{x\geq m} q_m(x) = 1$. Let $q'_m(x) = \frac{q_m(x)}{\sum_{x\geq m+1} q_m(x)}$. We have

$$\sup_{p\in\mathcal{P}} \left(\sum_{x\geq m} p(x)\log\frac{p(x)}{q(x)} + p(S_m)\log\frac{1}{p(S_m)} \right) \tag{4.2}$$

Where in 4.2 we use the fact that $\sum_{x\geq m} q_m(x) = 1$. Therefor, we can rewrite 4.2 as

$$\sup_{p\in\mathcal{P}} \Bigg(p(m)\log\frac{p(m)}{q_m(m)} + (p(S_m) - p(m))\log\frac{1}{1-q_m(m)} + \sum_{x\geq m+1} p(x)\log\frac{p(x)}{q'(x)} \\ + p(S_m)\log\frac{1}{p(S_m)}\Bigg)$$

$$= \sup_{p\in\mathcal{P}} \Bigg(p(S_m)\bigg(\frac{p(m)}{p(S_m)}\log\frac{p(m)/p(S_m)}{q_m(m)} + (1-\frac{p(m)}{p(S_m)})\log\frac{(1-\frac{p(m)}{p(S_m)})}{1-q_m(m)}\bigg) + p(m)\log p(S_m) \\ + (p(S_m)-p(m))\log\frac{p(S_m)}{p(S_m)-p(m)} + \sum_{x\geq m+1} p(x)\log\frac{p(x)}{q'(x)} + p(S_m)\log\frac{1}{p(S_m)}\Bigg)$$

But we can consider first two terms in the last equation as KL divergence between two Bernouli distributions $\mathcal{B}(p(m)/p(S_m))$ and $\mathcal{B}(q_m(m))$. Then

$$= \sup_p \left[p(S_m) D\bigg(\frac{p(m)}{p(S_m)}||q_m(m)\bigg) + \sum_{x\geq m+1} p(x)\log\frac{p(x)}{q'(x)} + p(S_{m+1})\log\frac{1}{p(S_{m+1})}\right] \\ \geq \tilde{\mathcal{T}}_{m+1}(\mathcal{P}).$$

□

Corollary 12. For all classes $\mathcal{P}$, $\lim_{m\to\infty} \tilde{\mathcal{T}}_m$ exists. □

Lemma 13. The limit $\lim_{m\to\infty} \mathcal{T}_m(\mathcal{P})$ exists and hence $\mathcal{T}(\mathcal{P}) = \lim_{m\to\infty} \mathcal{T}_m(\mathcal{P})$.

Proof If a collection $\mathcal{P}$ is not tight then for all m

$$\mathcal{T}_m(\mathcal{P}) = \infty. \tag{4.3}$$

To see this, suppose $\mathcal{T}_m(\mathcal{P}) < \infty$ for some m. Then there exists a distribution q_m and some $M < \infty$ such that

$$\sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x)\log\frac{p(x)}{q_m(x)} = M.$$

Consider the distribution q_1 that assigns probability $1/(m-1)$ for all numbers from 1 through

$m-1$. Then the distribution $q = (q_1 + q_m)/2$ satisfies

$$\sup_{p\in\mathcal{P}} \sum_{x\in\mathbb{N}} p(x) \log \frac{p(x)}{q(x)} = M + \log m + 1,$$

a contradiction of Lemma 6. We conclude then that $\mathcal{T}_m(\mathcal{P}) = \infty$ for all m, and the limit vacuously exists.

Therefore, we suppose in the rest of the proof that $\mathcal{P}$ is tight. Observe that

$$\mathcal{T}_m(\mathcal{P}) \le \tilde{\mathcal{T}}_m(\mathcal{P}).$$

Let $S_m = \{x \ge m\}$ as before and let q be any distribution over $\mathbb{N}$. Then

$$\begin{aligned}
&\sup_{p\in\mathcal{P}} \sum_{x\ge m} p(x) \log \frac{p(x)}{q(x)} \\
&= \sup_{p\in\mathcal{P}} \left(\sum_{x>m} p(x) \log \frac{p(x)}{q(x)} + p(S_m) \log \frac{p(S_m)}{p(S_m)} \right) \\
&\ge \sup_{p\in\mathcal{P}} \left(\sum_{x>m} p(x) \log \frac{p(x)}{q(x)} + p(S_m) \log \frac{1}{p(S_m)} + \inf_{\hat{p}\in\mathcal{P}} \hat{p}(S_m) \log \hat{p}(S_m) \right) \\
&\ge \inf_{q'} \sup_{p\in\mathcal{P}} \left(\sum_{x>m} p(x) \log \frac{p(x)}{q'(x)} + p(S_m) \log \frac{1}{p(S_m)} \right) + \inf_{\hat{p}\in\mathcal{P}} \hat{p}(S_m) \log \hat{p}(S_m) \\
&= \tilde{\mathcal{T}}_m(\mathcal{P}) + \inf_{\hat{p}\in\mathcal{P}} \hat{p}(S_m) \log \hat{p}(S_m)
\end{aligned}$$

Since $\mathcal{P}$ is tight, $\inf_{\hat{p}\in\mathcal{P}} \hat{p}(S_m) \log \hat{p}(S_m)$ goes to zero as $m \to \infty$. From Corollary 12, we know that the sequence $\{\tilde{\mathcal{T}}_m(\mathcal{P})\}$ has a limit. Therefore, the sequence $\mathcal{T}_m(\mathcal{P})$ also has a limit and in particular we conclude

$$\tilde{\mathcal{T}} = \lim_{m\to\infty} \mathcal{T}_m(\mathcal{P}). \qquad \square$$

Therefore, taking into account the above lemma, we can rephrase the definition of tail redundancy as in (4.1),

$$\mathcal{T}(\mathcal{P}) \overset{\text{def}}{=} \lim_{m\to\infty} \inf_{q_m} \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x) \log \frac{p(x)}{q_m(x)}$$

We now show that

$$\mathcal{T}(\mathcal{P}) = \min_q \lim_{m\to\infty} \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x) \log \frac{p(x)}{q(x)}.$$

Note that the limit above need not exist for every q. We take the above equation to mean the minimization over all q such that the limit exists. If no such q exists, the term on the right is considered to be vacuously infinite.

Lemma 14. For a collection $\mathcal{P}$ of distributions over $\mathbb{N}$ with tail redundancy $\mathcal{T}(\mathcal{P})$, there is a distribution q^* over $\mathbb{N}$ that satisfies

$$\lim_{m\to\infty} \sup_{p\in\mathcal{P}} \sum_{x>m} p(x) \log \frac{p(x)}{q^*(x)} = \mathcal{T}(\mathcal{P})$$

Proof If $\mathcal{P}$ is not tight, the lemma is vacuously true and any q is a "minimizer".

Therefore, we suppose in the rest of the proof that $\mathcal{P}$ is tight. From Lemma 6, we can pick a finite number m_r such that

$$\sup_{p\in\mathcal{P}} p(x > m_r) \leq \frac{1}{2^r},$$

and let q_r be any distribution that satisfies

$$\sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x) \log \frac{p(x)}{q_r(x)} \leq \arg\inf_q \sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x) \log \frac{p(x)}{q(x)} + \frac{1}{r}.$$

We then have

$$\lim_{r\to\infty} \sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x) \log \frac{p(x)}{q_r(x)} = \mathcal{T}(\mathcal{P}).$$

Take

$$q^*(x) = \sum_{r\geq 1} \frac{q_r(x)}{r(r+1)}.$$

Now we also have for $r \geq 4$ and any $m_r < m < m_{r+1}$ that

$$\begin{aligned}
\sup_{p\in\mathcal{P}} \sum_{x\geq m_r} p(x)\log\frac{p(x)}{q^*(x)} &\geq \sup_{p\in\mathcal{P}}\left(p(m_r \leq x < m)\log\frac{p(m_r \leq x < m)}{q^*(m_r \leq x < m)} + \sum_{x\geq m} p(x)\log\frac{p(x)}{q^*(x)}\right)\\
&\geq \sup_{p\in\mathcal{P}}\left(p(m_r \leq x < m)\log p(m_r \leq x < m) + \sum_{x\geq m} p(x)\log\frac{p(x)}{q^*(x)}\right)\\
&\geq -\frac{r}{2^r} + \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x)\log\frac{p(x)}{q^*(x)}.
\end{aligned}$$

as well as

$$\begin{aligned}
\sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x)\log\frac{p(x)}{q^*(x)} &\geq \sup_{p\in\mathcal{P}}\left(p(m \leq x < m_{r+1})\log\frac{p(m \leq x < m_{r+1})}{q^*(m \leq x < m_{r+1})} + \sum_{x\geq m_{r+1}} p(x)\log\frac{p(x)}{q^*(x)}\right)\\
&\geq \sup_{p\in\mathcal{P}}\left(p(m \leq x < m_{r+1})\log p(m \leq x < m_{r+1}) + \sum_{x\geq m_{r+1}} p(x)\log\frac{p(x)}{q^*(x)}\right)\\
&\geq -\frac{r+1}{2^{r+1}} + \sup_{p\in\mathcal{P}} \sum_{x\geq m_{r+1}} p(x)\log\frac{p(x)}{q^*(x)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\limsup_{m\to\infty} \sup_{p\in\mathcal{P}} \sum_{x>m} p(x)\log\frac{p(x)}{q^*(x)}\\
&\leq \lim_{r\to\infty}\left[\sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x)\log\frac{p(x)}{q_r(x)} + \frac{r}{2^r} + \frac{\log r(r+1)}{2^r}\right]\\
&= \mathcal{T}(\mathcal{P}).
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\liminf_{m\to\infty} \sup_{p\in\mathcal{P}} \sum_{x>m} p(x)\log\frac{p(x)}{q^*(x)}\\
&\geq \lim_{r\to\infty}\left[\sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x)\log\frac{p(x)}{q^*(x)} - \frac{r}{2^r}\right]\\
&\geq \lim_{r\to\infty}\left[\inf_{q}\sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x)\log\frac{p(x)}{q(x)} - \frac{r}{2^r}\right]\\
&\geq \lim_{r\to\infty}\left[\sup_{p\in\mathcal{P}} \sum_{x>m_r} p(x)\log\frac{p(x)}{q_r(x)} - \frac{1}{r} - \frac{r}{2^r}\right]\\
&= \mathcal{T}(\mathcal{P}),
\end{aligned}$$

and the lemma follows. □

Henceforth, we will describe any distribution q that achieves the minimizer in the lemma above as "q achieves the tail redundancy for $\mathcal{P}$".

Corollary 15. If a collection $\mathcal{P}$ of distributions is tight and has tail redundancy $\mathcal{T}(\mathcal{P})$, then there is a distribution q^* over $\mathbb{N}$ that satisfies

$$\lim_{m\to\infty} \sup_{p\in\mathcal{P}} \sum_{x>m} p(x)\log\frac{p(x)/\tau_p}{q^*(x)} = \mathcal{T}(\mathcal{P})$$

Proof The result follows using Lemma 14 and the fact that $\mathcal{P}$ is tight. □

4.1.2 Properties of the Tail Redundancy

We examine two properties of tail redundancy in this subsection. Note that the tail redundancy is the limit of $\mathcal{T}_m(\mathcal{P})$ as $m \to \infty$, but that $\mathcal{T}_m(\mathcal{P})$ need not always be negative. Therefore, we first assert that the limit, the tail redundancy, is always non-negative. The second concerns the behavior of tail redundancy across finite unions of classes. This property, while interesting inherently, also helps us cleanly characterize the per-symbol redundancy of *i.i.d.* sources in Section 4.2.

Lemma 16. For all $\mathcal{P}$, $\mathcal{T}(\mathcal{P}) \geq 0$.

Proof Again, if $\mathcal{P}$ is not tight, the lemma is trivially true from (4.3). Consider therefore the case where $\mathcal{P}$ is tight. Fix $m \in \mathbb{N}$, and let $S_m = \{x : x \geq m\}$. Then,

$$\inf_{q_m} \sup_{p \in \mathcal{P}} \sum_{x \geq m} p(x) \log \frac{p(x)}{q(x)} \geq \sup_{p \in \mathcal{P}} \left\{ p(S_m) \log p(S_m) \right\}$$

Furthermore, $\sup_p p(S_m) \log p(S_m) \to 0$ as $m \to \infty$ because $\sup_p p(S_m) \to 0$ as $m \to \infty$. To see this, note that if $\sup_p p(S_m) \leq \frac{1}{e}$, then

$$\sup_p p(S_m) \log p(S_m) \geq \left(\sup_{p'} p'(S_m) \right) \log \left(\sup_{p'} p'(S_m) \right).$$

The lemma follows. □

Lemma 17. Let $\{a_i^{(j)}\}$, $1 \leq j \leq k$ be k different sequences with limits $a^{(j)}$ respectively. For all i, let

$$\hat{a}_i = \max_j a_i^{(j)}.$$

Then the sequence $\{\hat{a}_i\}$ has a limit and the limit equals $\max a^{(j)}$.

Proof Wolog, let the sequences be such that the limits are $a^{(1)} \geq a^{(2)} \geq \ldots \geq a^{(k)}$. Consider any $0 < \epsilon < \frac{a^{(1)} - a^{(2)}}{2}$. Then for all $1 \leq j \leq k$, there exist N_j such that for all $i \geq N_j$, $|a_i^{(j)} - a^{(j)}| \leq \epsilon$. Let $N = \max N_j$. We now have that for all $i \geq N$,

$$\hat{a}_i = \max_j a_i^{(j)} = a_i^{(1)},$$

and therefore, the sequence $\{\hat{a}_i\}$ has a limit, and is equal to $a^{(1)} = \max_{1 \leq j \leq k} \lim_{i \to \infty} a_i^{(j)}$. □

Lemma 18. Let $\mathcal{T}(\mathcal{P}_1), \mathcal{T}(\mathcal{P}_2), \ldots, \mathcal{T}(\mathcal{P}_k)$ be tail redundancy of collections $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$ respectively. Then

$$\mathcal{T}(\cup_{i=1}^k \mathcal{P}_i) = \max_{1 \leq j \leq k} \mathcal{T}(\mathcal{P}_j).$$

Proof We first observe $\mathcal{T}(\cup_{i=1}^k \mathcal{P}_i) \geq \max_{1\leq j\leq k} \mathcal{T}(\mathcal{P}_j)$, since for all q, and all $1 \leq j \leq k$, we have

$$\begin{aligned}\sup_{p\in\cup_i\mathcal{P}_i} \sum_{x\geq m} p(x)\log\frac{p(x)}{q(x)} &\geq \sup_{p\in\mathcal{P}_j} \sum_{x\geq m} p(x)\log\frac{p(x)}{q(x)}\\ &\geq \inf_{q'} \sup_{p\in\mathcal{P}_j} \sum_{x\geq m} p(x)\log\frac{p(x)}{q'(x)}\\ &= \mathcal{T}_m(\mathcal{P}_j)\end{aligned}$$

To show that $\mathcal{T}(\cup_{i=1}^k \mathcal{P}_j) \leq \max_j \mathcal{T}(\mathcal{P}_j)$, let $q_1, q_2, \ldots, q_k$ be distributions that achieve the tail redundancies $\mathcal{T}(\mathcal{P}_1), \mathcal{T}(\mathcal{P}_2), \ldots, \mathcal{T}(\mathcal{P}_k)$, respectively and let $\hat{q}(x) = \frac{\sum_{i=1}^k q_i(x)}{k}$ for all $x \in \mathcal{X}$. Furthermore, for all distributions q over naturals and collections of distributions $\mathcal{P}$, let

$$\mathcal{T}_m(\mathcal{P}, q) = \sup_{p\in\mathcal{P}} \sum_{x\geq m} p(x)\log\frac{p(x)}{q(x)}.$$

Clearly, we have

$$\mathcal{T}(\cup_{i=1}^k \mathcal{P}_i) = \lim_{m\to\infty} \inf_q \mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, q).$$

We will attempt to understand the behaviour of the sequence $\mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, \hat{q})$ first. Observe that

$$\begin{aligned}\mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, \hat{q}) &= \max_{1\leq j\leq k} \sup_{p\in\mathcal{P}_j} \sum_{x\geq m} p(x)\log\frac{p(x)}{\hat{q}(x)}\\ &= \max_{1\leq j\leq k} \mathcal{T}_m(\mathcal{P}_j, \hat{q}). \qquad (4.4)\end{aligned}$$

The limit $\lim_{m\to\infty} \mathcal{T}_m(\mathcal{P}_j, \hat{q})$ exists and is equal to $\mathcal{T}(\mathcal{P}_j)$. This follows because

$$\begin{aligned}\mathcal{T}_m(\mathcal{P}_j, \hat{q}) &\overset{(a)}{\leq} \sup_{p\in\mathcal{P}_j} \left(\sum_{x\geq m} p(x)\log\frac{p(x)}{q_1(x)} + \sum_{x\geq m} p(x)\log k\right)\\ &\leq \sup_{p\in\mathcal{P}_j} \left(\sum_{x\geq m} p(x)\log\frac{p(x)}{q_1(x)}\right) + \sup_{p\in\mathcal{P}_j} \sum_{x\geq m} p(x)\log k,\end{aligned}$$

where (a) follows because for all x, $q(x) \geq q_1(x)/k$. Let $\delta_m = \sup_{p\in\mathcal{P}_j} p(x)$. Note that since $\mathcal{P}_j$ is tight, we have $\lim_{m\to\infty} \delta_m = 0$. Thus

$$\inf_q \sup_{p\in\mathcal{P}_j} \sum_{x\geq m} p(x) \log \frac{p(x)}{q(x)} \leq \mathcal{T}_m(\mathcal{P}_j, \hat{q}) \leq \sup_{p\in\mathcal{P}_j} \left(\sum_{x\geq m} p(x) \log \frac{p(x)}{q_1(x)} \right) + \delta_m \log k$$

and both the lower and upper bound on $\mathcal{T}_m(\mathcal{P}_j, q)$ are sequences whose limit exists, and both limits are $\mathcal{T}(\mathcal{P}_j)$. We conclude then, that for all $1 \leq j \leq k$,

$$\lim_{m\to\infty} \mathcal{T}_m(\mathcal{P}_j, \hat{q}) = \mathcal{T}(\mathcal{P}_j). \tag{4.5}$$

From (4.4), (4.5) and Lemma 17, we have that the sequence $\mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, \hat{q})$ also has a limit and that

$$\lim_{m\to\infty} \mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, \hat{q}) = \max_{1\leq j\leq k} \mathcal{T}(\mathcal{P}_j).$$

Putting it all together, we have

$$\mathcal{T}(\cup_{i=1}^k \mathcal{P}_i) = \lim_{m\to\infty} \inf_q \mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, q) \leq \lim_{m\to\infty} \mathcal{T}_m(\cup_{i=1}^k \mathcal{P}_i, \hat{q}) = \max_{1\leq j\leq k} \mathcal{T}(\mathcal{P}_j).$$

The lemma follows. □

4.2 Main Result

In [19] we showed that if a collection of distributions has finite single letter redundancy, then a couple of technical conditions, one of which was similar to but not the same as the the tail redundancy condition being 0, then the collection was strongly compressible. At the same time, we also had noted that the technical conditions therein were not necessary. The main result of this section is that the per-symbol redundancy goes to tail redundancy as n increase. In fact, we first show that per-symbol redundancy is always greater than or equal to tail redundancy $n \to \infty$ and conversely we show that tail redundancy is always greater than or equal per-symbol redundancy as $n \to \infty$. This result implies that zero tail redundancy is a necessary and sufficient condition for a collection to be strongly compressible.

Lemma 19. Consider collection $\mathcal{P}$. if single letter redundancy $R_1(\mathcal{P}) = \infty$ then $\mathcal{T}(\mathcal{P}) = \infty$.

Proof Assume on the contrary that $\mathcal{T}(\mathcal{P})$ is finite, then for any m,

$$\inf_q \sup_p \sum_{x \geq m} p(x) \log \frac{p(x)}{q(x)} = M < \infty.$$

Let q_m be the encoder that achieves M, i.e.

$$\sup_p \sum_{x \geq m} p(x) \log \frac{p(x)}{q_m(x)} = M.$$

Consider $q_0 = (\frac{1}{m}, \ldots, \frac{1}{m})$, a uniform distribution over $\{1, 2, 3, \ldots, m\}$, and let $q(x) = \frac{q_0(x)+q_m(x)}{2}$. Then for all $p \in \mathcal{P}$,

$$R_1 = \sum_{x=1}^{\infty} p(x) \log \frac{p(x)}{q(x)} \leq \log 2m + M + 1 < \infty,$$

which is a contradiction. □

Theorem 20. Let $\mathcal{P}$ be a collection of distributions over $\mathbb{N}$ and $\mathcal{P}^\infty$ be the collection of all measures over infinite sequences that can be obtained by *i.i.d.* sampling from a distribution in $\mathcal{P}$. Then

$$\lim_{n \to \infty} \frac{1}{n} R_n(\mathcal{P}^\infty) = \mathcal{T}(\mathcal{P}).$$

□

A couple of quick examples first.

Example 5. Proposition 2 proved that $\mathcal{B}^\infty$ is not strongly compressible, and we note that $\mathcal{T}(\mathcal{B}) > 0$. □

Example 6. Let $h > 0$. Let $\mathcal{M}_h$ be the collection of uniform distributions over $\mathbb{N}$ such that

$$E_p\left(\log\frac{1}{p(X)}\right)^2 < h.$$

Let $\mathcal{M}_h^\infty$ be the set of all *i.i.d.* distributions with one dimensional marginals from $\mathcal{M}_h$. Then it is easy to verify that $\mathcal{T}(\mathcal{M}_h) = 0$ and that $\mathcal{M}_h^\infty$ is strongly compressible. Specifically, we can construct a measure q^* over infinite sequences of naturals whose per-symbol length-n redundancy against sources in $\mathcal{M}_h^\infty$ is upper bounded by (see [26])

$$\frac{2h^{3/2}}{\sqrt{\log n}} + \pi\sqrt{\frac{2}{3n}}$$

so $\mathcal{M}_h^\infty$ is strongly compressible. □

4.3 Proof of Theorem 20

If $\mathcal{P}$ is not tight then using Lemma 6, $R_1 = \infty$ and using Lemma 19, the tail redundancy $\mathcal{T}(\mathcal{P})$ is infinite. Also, $R_1 = \infty$ results in $\frac{1}{m}R_m(\mathcal{P}^\infty) = \infty$, so if $\mathcal{P}$ is not tight, $\frac{1}{m}R_m(\mathcal{P}^\infty) = \mathcal{T}(\mathcal{P}) = \infty$. If $\mathcal{P}$ is tight we first show that $\frac{1}{m}R_m(\mathcal{P}^\infty) \geq \mathcal{T}(\mathcal{P})$ and then we show that $\frac{1}{m}R_m(\mathcal{P}^\infty) \leq \mathcal{T}(\mathcal{P})$.

4.3.1 Direct Part

In this section, we show the direct part of theorem 20, i.e. we prove that

$$\lim_{n\to\infty}\frac{1}{n}R_n(\mathcal{P}^\infty) \geq \mathcal{T}(\mathcal{P}).$$

If $\mathcal{P}$ is tight, then for any $c > 0$, we can find a finite number m_n such that $\forall p \in \mathcal{P}$,

$$p(x \geq m_n) < \frac{c}{n}.$$

Let $\tau_n^p \overset{\text{def}}{=} p(x \geq m_n)$ be the tail probability past m_n under p. For each sequence x^n, let the auxiliary sequence y^n be defined by

$$y_i = \begin{cases} x_i & \text{if} \quad x_i \leq m_n \\ -1 & \text{if} \quad x_i \geq m_n. \end{cases}$$

Let $\mathcal{Y} = \{-1, 1, 2, \ldots, m_n\}$, then $y^n \in \mathcal{Y}^n$. For $n \geq 1$, let r_{X^n} be distributions over length$-n$ strings of natural numbers. We show that $\forall n$, and for all r_{X^n},

$$\sum p(x^n) \log \frac{p(x^n)}{r_{X^n}} \geq \mathcal{T}(\mathcal{P}).$$

Proving also that

$$\lim_{n\to\infty} \inf_{q} \sup_{p\in\mathcal{P}^\infty} \sum p(x^n) \log \frac{p(x^n)}{r_{X^n}} \geq \mathcal{T}(\mathcal{P}).$$

Now

$$r(x^n) = r(x^n, y^n) = r(x^n|y^n) r(y^n),$$

and

$$\begin{aligned}
&\sup_{p\in\mathcal{P}^n} D(p_{X^n}||r_{X^n}) \\
&= \sup_{p\in\mathcal{P}^n} \sum_{X^n} p(X^n) \log \frac{p(X^n)}{r(X^n)} \\
&= \sup_{p\in\mathcal{P}^n} \Bigg(\sum_{y^n\in\mathcal{Y}^n} p(y^n) \sum_{X^n} p(X^n|y^n) \log \frac{p(X^n|y^n)}{r(X^n|y^n)} \\
&\qquad + \sum_{y^n\in\mathcal{Y}^n} p(y^n) \sum_{X^n} p(X^n|y^n) \log \frac{p(y^n)}{r(y^n)} \Bigg) \\
&\overset{(a)}{\geq} \sup_{p\in\mathcal{P}^n} \sum_{y^n\in G} p(y^n) \sum_{X^n} p(X^n|y^n) \log \frac{p(X^n|y^n)}{r(X^n|y^n)}.
\end{aligned}$$

Where (a) is using the fact that KL divergence is greater than or equal to 0 and $G = \{y^n : \text{only one element of } y^n \text{ is} -1\}$. Note that for a given $y^n \in G$, it is easy to see that $p(x|x \geq m_n) = p(X^n|y^n)$. Also, we can characterize a single letter encoder $r(_{y^n}(x)$ from

$r(X^n|y^n)$. Therefore,

$$\begin{aligned}
&\sup_{p\in\mathcal{P}^n} \sum_{y^n\in G} p(y^n) \sum_{X^n} p(X^n|y^n) \log \frac{p(X^n|y^n)}{r(X^n|y^n)} \\
&\geq \sup_{p\in\mathcal{P}} \sum_{y^n\in G} p(y^n) \sum_{x\geq m_n} p(x|x\geq m_n) \log \frac{p(x|x\geq m_n)}{r(x|y^n)} \\
&= \sup_{p\in\mathcal{P}} \sum_{y^n\in G} \frac{p(y^n)}{\tau_n^p} \sum_{x>m_n} p(x) \log \frac{p(x)/\tau_n^p}{r_{y^n}(x)}.
\end{aligned}$$

For all $p\in\mathcal{P}$, let

$$\bar{y}^*(p) = \arg\min_{y^n\in G} \sum_{x\geq m_n} p(x) \log \frac{p(x)}{r_{y^n}(x)}.$$

Then,

$$\sup_p \sum_{y^n\in G} \frac{p(y^n)}{\tau_n^p} \sum_{x>m_n} p(x) \log \frac{p(x)/\tau_n^p}{r_{y^n}(x)} \geq \sup_p \frac{p(G)}{\tau_n^p} \sum_{x\geq m_n} p(x) \log \frac{p(x)/\tau_n^p}{r_{\bar{y}^*(p)}(x)}.$$

From the fact that $\forall p\in\mathcal{P}$, $p(G) = n(1-\tau_n^p)^{n-1}\tau_n^p$, we therefore have

$$\begin{aligned}
\sup_p D(p_{X^n}||r_{X^n}) &\geq \sup_p \frac{p(G)}{\tau_n^p} \sum_{x\geq m_n} p(x) \log \frac{p(x)/\tau_p^n}{r_{\bar{y}^*(p)}(x)} \\
&\geq \sup_p \frac{p(G)}{\tau_n^p} \left(\sum_{x\geq m_n} p(x) \log \frac{p(x)}{r_{\bar{y}^*(p)}(x)} + \tau_n^p \log \frac{1}{\tau_n^p} \right) \\
&\geq \sup_p n(1-\frac{c}{n})^n \left(\sum_{x\geq m_n} p(x) \log \frac{p(x)}{r_{\bar{y}^*(p)}(x)} + \tau_n^p \log \frac{1}{\tau_n^p} \right). \qquad (4.6)
\end{aligned}$$

Let $\mathcal{P}_{y^n} = \{p\in\mathcal{P} : \bar{y}^*(p) = y^n\}$. Then $\mathcal{P} = \cup_{y^n}\mathcal{P}_{y^n}$. Note that $\cup_{y^n}\mathcal{P}_{y^n}$ is a finite union, therefore there exists $\bar{y}$ such that $\mathcal{T}(\mathcal{P}_{\bar{y}}) = \mathcal{T}(\mathcal{P})$.

Then

$$\begin{aligned}
\sup_{p\in\mathcal{P}}\left[\sum_{x\geq m_n} p(x)\log\frac{p(x)}{r_{y^*(p)}(x)} + \tau_n^p \log\frac{1}{\tau_n^p}\right] &= \max_{y^n}\sup_{p\in\mathcal{P}_{y^n}}\left[\sum_{x\geq m_n} p(x)\log\frac{p(x)}{r_y(x)} + \tau_n^p\log\frac{1}{\tau_n^p}\right] \\
&\geq \max_{y^n}\left(\inf_q \sup_{p\in\mathcal{P}_{y^n}}\left[\sum_{x\geq m_n} p(x)\log\frac{p(x)}{r_{y^n}(x)} + \tau_n^p\log\frac{1}{\tau_n^p}\right]\right) \\
&\geq \max_{y^n}\tilde{\mathcal{T}}_{m_n}(\mathcal{P}_{y^n}) \\
&\overset{(*)}{\geq} \max_{y^n}\mathcal{T}(\mathcal{P}_{y^n}) \\
&= \mathcal{T}(\mathcal{P}), \qquad (4.7)
\end{aligned}$$

where $(*)$ follows since $\tilde{\mathcal{T}}_m$ monotonically decreases to the limit $\mathcal{T}$. Putting (4.6) and (4.7) together, we obtain

$$\sup_{p\in\mathcal{P}} D(p(_{X^n}||r_{X^n}) \geq n(1-\frac{c}{n})^n\mathcal{T}(\mathcal{P}).$$

Since the inequality holds for all $c > 0$, we can keep c small enough , so that

$$\sup_{p\in\mathcal{P}}\frac{1}{n}D(p(_{X^n}||r_{X^n}) \geq \mathcal{T}(\mathcal{P}).$$

4.3.2 Converse Part

We now prove the converse part of Theorem 20, i.e. we show that

$$\lim_{n\to\infty}\frac{1}{n}R_n(\mathcal{P}^n) \leq \mathcal{T}(\mathcal{P}).$$

Let r_n be an optimal universal encoder for any finite m-ary alphabet, length-n sequences. Let the redundancy of r_n against $(m+1)-$ary *i.i.d.* sequences of length n be S_m. It is known that $S_m \sim \frac{m}{2}\log n$ [2–4].

Let q^* be the distribution that achieves $\mathcal{T}(\mathcal{P})$. Like we will see later, we set $m = \sqrt{n}$. As before, we construct an auxiliary sequence y^n from x^n where

$$y_i = \begin{cases} x_i & \text{if} \quad x_i \le m \\ -1 & \text{if} \quad x_i \ge m. \end{cases}$$

Given any sequence $y^n \in \{-1, [m]\}^n$, and $x^n \in \mathbb{N}^n$, we say $x^n \sim y^n$ if y^n is consistent with x^n (constructed from x^n). Note that without loss of generality we can assume $\sum_{x \ge m} q^*(x) = 1$ (otherwise we can construct another encoder from that with smaller redundancy). Then, construct q by first encoding the auxiliary sequence y^n

$$q(y^n) = r(y^n),$$

followed by describing x_i for each y_i where $y_i = -1$ independently,

$$q(x^n|y^n) = \prod_{i=1:n} q(x_i|y_i),$$

where to describe x_i when $y_i = -1$ we use the distribution q^*

$$q(x_i|y_i) = \begin{cases} q^*(x_i) & \text{if} \quad y_i = -1 \\ 1 & \text{if } y_i \ne -1, x_i = y_i \\ 0 & \text{if } y_i \ne -1, x_i \ne y_i. \end{cases}$$

Then,

$$\begin{aligned} &\frac{1}{n} \sum_{x^n \in \mathbb{N}^n} p(x^n) \log \frac{p(x^n)}{q(x^n)} \\ &= \frac{1}{n} \sum_{y^n \in \mathbb{N}^m} p(y^n) \log \frac{p(y^n)}{q(y^n)} + \frac{1}{n} \sum_{x^n \in \mathbb{N}^n} p(x^n) \log \frac{p(x^n|y^n)}{q(x^n|y^n)} \\ &\le \frac{S_m}{n} + \frac{1}{n} \sum_{y^n \in \mathbb{N}^m} p(y^n) \sum_{x^n \in \mathbb{N}^n} p(x^n|y^n) \log \frac{p(x^n|y^n)}{q(x^n|y^n)}. \end{aligned} \tag{4.8}$$

Fix y^n and let $k(y^n)$ be the number of -1 in y^n. Let $\tau_p = p(x \ge m)$, then for a fixed y^n,

$\forall x^n \sim y^n$,

$$p(x^n|y^n) = \prod_{i:y_i=-1} \frac{p(x_i)}{\tau_p}.$$

We can rewrite the second term in equation (4.8) as

$$\begin{aligned}\frac{1}{n}\sum_{y^n\in\mathbb{N}^m} p(y^n) \sum_{\substack{x^n\in\mathbb{N}^n\\ x^n\sim y^n}} p(x^n|y^n)\log\frac{p(x^n|y^n)}{q(x^n|y^n)} &= \frac{1}{n}\sum_{y^n\in\mathbb{N}^m} p(y^n) \sum_{\substack{x^n\in\mathbb{N}^n\\ x^n\sim y^n}} \prod_{j:y_j=-1}\frac{p(x_j)}{\tau_p}\log\prod_{i:y_i=-1}\frac{p(x_i)/\tau_p}{w(x_i)}\\ &= \frac{1}{n}\sum p(y^n)A(k(y^n)).\end{aligned}$$

We can bound $A(k(y^n))$ as follows,

$$A(k(y^n)) = \sum_{j:y_j=-1}\sum_{x_j\geq m}\frac{p(x_j)}{\tau_p}\log\frac{p(x_j)/\tau_p}{q^*(x_j)} \leq k(y^n)\sum_{x\geq m}\frac{p(x)}{\tau_p}\log\frac{p(x)/\tau_p}{q^*(x)}.$$

Now we have,

$$\begin{aligned}\sum_{y^n} p(y^n)A(k(y^n)) &\leq \sum_{y^n} p(y^n)k(y^n)\sum_{x\geq m}\frac{p(x)}{\tau_p}\log\frac{p(x)/\tau_p}{q^*(x)}\\ &= \sum_{x\geq m}\frac{p(x)}{\tau_p}\log\frac{p(x)/\tau_p}{q^*(x)}\sum_{y^n} p(y^n)k(y^n)\\ &= E(\#y_i=-1)\sum_{x\geq m}\frac{p(x)}{\tau_p}\log\frac{p(x)/\tau_p}{q^*(x)}. \qquad (4.9)\end{aligned}$$

Combining equation (4.8) and (4.9), we have

$$\begin{aligned}\frac{1}{n}\sum_{x^n} p(x^n)\log\frac{p(x^n)}{q(x^n)} &\leq \frac{S_m}{n} + \frac{1}{n}E(\#y_i=-1)\Bigg(\sum_{x\geq m}\frac{p(x)}{\tau_p}\log\frac{p(x)/\tau_p}{q^*(x)}\Bigg)\\ &\leq \frac{S_m}{n} + \sum_{x\geq m} p(x)\log\frac{p(x)/\tau_p}{q^*(x)},\end{aligned}$$

where the last inequality follows since $E(\#y_i=-1) = n\tau_p$. Now, taking the supremum over all p and the limit, we have

$$\lim_{n\to\infty} \sup_{p\in\mathcal{P}^n} \frac{1}{n} \sum_{x^n\in\mathbb{N}^n} p(x^n) \log \frac{p(x^n)}{q(x^n)} \leq \lim_{n\to\infty} \sup_{p\in\mathcal{P}} \left[\frac{S_m}{n} + \sum_{x\geq m} p(x) \log \frac{p(x)/\tau_p}{q^*(x)} \right].$$

Since $m = \sqrt{n}$, as n goes to infinity $\frac{S_m}{n}$ goes to 0 and using Corollary 15 the second term goes to $\mathcal{T}(\mathcal{P})$. Therefore,

$$\lim_{n\to\infty} \frac{1}{n} R_n(\mathcal{P}^n) \leq \mathcal{T}(\mathcal{P}).$$

5

Unbounded Memory Markov Sources

In this chapter, we study the redundancy of universally compressing strings $X_1, \ldots, X_n$ generated by a binary Markov source p without any bound on the memory. To better understand the connection between compression and estimation in the Markov regime, we consider a class of Markov sources restricted by a *continuity* condition. In the absence of an upper bound on memory, the continuity condition implies that $p(X_0|X_{-m}^{-1})$ gets closer to the true probability $p(X_0|X_{-\infty}^{-1})$ as m increases, rather than vary around arbitrarily. For such sources, we prove asymptotically matching upper and lower bounds on the redundancy. In the process, we identify what sources in the class matter the most from a redundancy perspective.

5.1 Introduction

Estimation and compression are virtually synonymous in the *i.i.d.* regime. Indeed, in the *i.i.d.* case, the add-half (and other add-constant) estimators that provide reasonable estimates of probabilities of various symbols are described naturally using a universal compression setup. These estimators simply correspond to the conditional probabilities assigned by a Bayesian mixture when Dirichlet priors on the parameter space—and indeed encoding the probabilities given by these Bayesian mixtures yields good universal compression schemes for these classes of distributions.

In the Markov setup, there is an additional complication not seen in the *i.i.d.* setup—*mixing*—that complicates the relation between compression and estimation. Without going into the technicalities of mixing, slow mixing sources do not explore the state space efficiently.

For example, consider a memory-1 binary Markov source that assigns conditional probability of $1-\epsilon$ for a 1 given 1, and ϵ for the conditional probability of a 1 given 0. If we start the source from the state 1, for sample length $n \ll \frac{1}{\epsilon}$, we will see a sequence of all 1s with high probability in length-n samples. This sequence of 1s is, of course, easy to compress—but clearly precludes the estimation of the conditional probabilities associated with 0.

Previous work in the Markov regime, however, has typically considered classes Markov sources with bounded memory (say, all memory-3 Markov sources) as a natural hierarchy of classes. As the prior example shows, these classes are definitely not natural from an estimation point of view. Small memory sources—even with memory one can be arbitrarily slow mixing—and hence hard to estimate. On the other hand, sources with longer memory may be easier to estimate if they are fast mixing and satisfy certain other conditions, as we will see below.

As a consequence, we look for a different way to resolve the class of all finite memory Markov sources into a nested hierarchy of classes. Therefore, in [28], a new class of Markov sources was introduced, one that was more amenable to estimation. These classes of sources were not bounded in memory, rather they can have arbitrarily long memory. However, these sources satisfy a continuity condition [28, 29], as described technically in Section 5.2.1.

Roughly speaking, the continuity condition imposes two intuitive constraints closely related to each other. Let p be a binary Markov source with finite but unknown memory, and consider $p(X_0|X_{-\infty}^{-1})$. Because the source has finite memory, there is a suffix of the past, $X_{-\infty}^{-1}$ that determines the conditional probability above. Since we do not have an a-priori bound on the memory, we cannot say how much of the past we need. Yet, the conditional probabilities $p(X_0|X_{-m}^{-1})$ are well defined for all m. It is now possible to construct Markov sources where, unless we have suffixes long enough to encapsulate the true state (or equivalently, m larger than the true memory), $p(X_0|X_{-m}^{-1})$ is simply not a reflection of the true probability $p(X_0|X_{-\infty}^{-1})$.

The continuity condition prohibits this pathological property—imposing on the other hand that the more of the context (X_{-m}^{-1}) we see, the better $p(X_0|X_{-m}^{-1})$ reflects $p(X_0|X_{-\infty}^{-1})$. Second, given a history X_{-m}^{-1}, the continuity condition implies that the conditional information one more bit in the past, X_{-m-1}^{-1}, provides on X_0 diminishes with m.

The continuity condition may be imagined as a soft constraint on the memory, but it does not control mixing. Suppose we consider the collection of all Markov sources that satisfy a given continuity condition. It was shown in [28] that it is possible to use length-n samples to estimate the conditional probabilities $p(1|\mathbf{w})$ of all strings $\mathbf{w}$ of length $\log n$ that appear in the sample, as well as provide deviation bounds on the estimates.

This hints that Markov sources nested by the continuity condition (as opposed to memory) are a natural way to break down the collection of all Markov sources. In order to better understand these model classes, we study compressing Markov sources constrained by the continuity condition, and obtain asymptotically tight bounds on their redundancy.

Part of the reason to study this is to understand what portions of the model classes are more important (namely, contribute primarily) to the redundancy. Indeed, it turns out that the primary contribution to the redundancy comes from essentially fast mixing sources whose state probabilities that are not towards the extremes (not near 0 or 1), while, of course, these sources do not complicate estimation at all. On the other hand, slow mixing sources hit estimation, but do not matter for compression at all—a dichotomy which was hinted at with our very first memory-1 example.

Our main results are matching lower and upper bounds, in Sections 5.3 and 5.4 respectively. In Section 5.2, we set up notations, and briefly describe the continuity condition in 5.2.1.

5.1.1 Prior Work

Davisson formulated the average and worst case redundancy in his seminal paper in 1973 [2]. A long sequence of work has characterized the worst case redundancy for memoryless sources [30] [6] [31] [32]. Average redundancy of Markov sources with fixed memory has been studied in [33] [34] [35]. In [36], the authors obtain the worst case redundancy of such Markov sources and later [37] derived the exact worst case redundancy of such Markov sources. The estimation and compression of finite memory context tree models was studied in [38] and [39]. [29] and [40] studied the estimation of context tree with unbounded memory.

For a different comparison of estimation and compression in Markov settings, see [41]. Here the authors obtain the redundancy of conditionally describing one additional symbol after obtaining a sample. Finally, [42] considers compression of patterns of Hidden Markov models.

5.2 Setup and Notation

We denote length-n strings in bold face $\mathbf{x}$ or as x_1^n, and their random counterparts are $\mathcal{X}$ and X_1^n respectively. For any $\mathbf{x} \in \{0,1\}^n$ and $\mathbf{w} \in \{0,1\}^\ell$ with $\ell \leq n$, let $n_{\mathbf{w}}$ be the number of appearances of $\mathbf{w}$ in $\mathbf{x}$ as a *substring*. For $\mathbf{x}, \mathbf{y} \in \{0,1\}^*$, $\mathbf{x} \prec \mathbf{y}$ denotes that $\mathbf{x}$ is a suffix of $\mathbf{y}$ (*e.g.* $010 \prec 11010$), $\mathbf{xy}$ is the concatenation of $\mathbf{x}$ followed by $\mathbf{y}$ (*e.g.* if $\mathbf{x} = 010$ and $\mathbf{y} = 1$ then $\mathbf{xy} = \mathbf{x}1 = 0101$), and $|\mathbf{x}|$ as the length of $\mathbf{x}$.

We consider a two-sided infinite binary Markov random process p generating a sample $\cdots, X_{-1}, X_0, X_1, \cdots$. The memory of p is finite (though not bounded a-priori), and let S_p to be *context tree* [6] of source p. S_p is a set of leaves of a complete binary tree and p is completely described by the conditional (transition) probabilities $p(1 \mid \mathbf{s})$ for $\mathbf{s} \in S_p$.

Let $\mathcal{M}^\ell$ to be all the Markov chains with memory at most ℓ, and $\mathcal{M} = \cup_\ell \mathcal{M}^\ell$ be the family of all the finite memory Markov chains. As mentioned before, while $\mathcal{M}^\ell$ is a natural class, we are looking to break $\mathcal{M}$ down into a more natural hierarchy of classes.

5.2.1 Markov Chain with Continuity Condition

Let $\delta := \mathbb{N} \to \mathbb{R}^+$ be a function such that $\delta(n) \to 0$ as $n \to \infty$. A Markov chain p satisfies the *continuity condition* subject to δ if for all $\mathbf{s}_1, \mathbf{s}_2 \in S_p$, and $a \in \{0,1\}$, we have

$$\left| \frac{p(a \mid \mathbf{s}_1)}{p(a \mid \mathbf{s}_2)} - 1 \right| \le \delta(|\mathbf{w}|)$$

for all $\mathbf{w} \in \{0,1\}^*$ such that $\mathbf{w} \prec \mathbf{s}_1$ and $\mathbf{w} \prec \mathbf{s}_2$ (namely $\mathbf{w}$ is a common suffix of $\mathbf{s}_1$ and $\mathbf{s}_2$). For technical reasons, we will assume that $\delta(n) < \frac{1}{4n}$, see [28].

Denote $\mathcal{M}_\delta$ to be the family of all the Markov chains with continuity condition subject to δ and $\mathcal{M}_\delta^\ell = \mathcal{M}_\delta \cap \mathcal{M}^\ell$. Roughly speaking, the continuity condition constraints the transition probabilities of states with long common suffix to be close.

Let $S_p(\mathbf{w}) = \{\mathbf{s} \in S_p : \mathbf{w} \prec \mathbf{s}\}$. Clearly we have that the stationary probability $p(\mathbf{w}) = \sum_{s \in S_p(\mathbf{w})} p(\mathbf{s})$ and that $p(1|\mathbf{w}) = \Big(\sum_{s \in S_p(\mathbf{w})} p(1|\mathbf{s})p(\mathbf{s})\Big)/p(\mathbf{w})$. Let p have memory ℓ. In analogy to the true conditional probability $p(1|\mathbf{w})$, for a given $\mathbf{x} \in \{0,1\}^n$ and past $x_{-\ell+1}^0$, let

$$\tilde{p}(1 \mid \mathbf{w}) = \frac{\displaystyle\sum_{s \in S_p(\mathbf{w})} n_{\mathbf{s}} p(1 \mid \mathbf{s})}{n_{\mathbf{w}}}, \tag{5.1}$$

be the *empirical aggregated distribution* of p, write it as $\tilde{p}_{\mathbf{w}}$ for simplicity. In a slight abuse of notation here, $n_{\mathbf{s}}$ (respectively $n_{\mathbf{w}}$) is the number of bits in $\mathbf{x}$ with context $\mathbf{s}$ (respectively

$\mathbf{w}$), *i.e.,* number of bits in $\mathbf{x}$ immediately preceded by $\mathbf{s}$ (respectively $\mathbf{w}$) when taking the past $x_{-\ell+1}^0$ into account).

5.2.2 The Redundancy

For any distribution family $\mathcal{P}$ on $\{0,1\}^n$, the worst case minimax redundancy of $\mathcal{P}$ is defined as

$$R(\mathcal{P}) = \inf_q \sup_{p\in\mathcal{P}} \max_{\mathbf{x}\in\{0,1\}^n} \log \frac{p(\mathbf{x})}{q(\mathbf{x})},$$

similarly, the average minimax redundancy is defined as

$$\tilde{R}(\mathcal{P}) = \inf_q \sup_{p\in\mathcal{P}} E_{\mathcal{X}\sim p} \log \frac{p(\mathcal{X})}{q(\mathcal{X})},$$

where q is choosing from all the possible distributions on $\{0,1\}^n$. For any given (*fixed*) past $x_{-\infty}^0$, we know that for any $p \in \mathcal{M}$ we will have a well defined distribution over $\{0,1\}^n$, given by

$$\bar{p}(x_1^n) = p(x_1^n \mid x_{-\infty}^0).$$

The main result of this chapter is a lower and upper bound on the average redundancy of $\mathcal{M}_\delta$ over $\{0,1\}^n$ for any past, i.e.

$$\tilde{R}(\mathcal{M}_\delta) \triangleq \inf_q \sup_{p\in\mathcal{M}_\delta} \sup_{x_{-\infty}^0} E_{\mathcal{X}\sim\bar{p}} \log \frac{\bar{p}(\mathcal{X})}{q(\mathcal{X})}.$$

5.3 The Lower Bound

The *Redundancy-Capacity* theorem [43] is a common approach to lower bound the minimax redundancy. This approach gets complicated in our case since there is no universal bound on the memory of sources in $\mathcal{M}_\delta$, rendering the parameter space to be infinite dimensional. We therefore first consider $\mathcal{M}_\delta^\ell$ (see Section II), the subset of sources in $\mathcal{M}_\delta$ which also have finite memory $\leq \ell$, and obtain a lower bound on $\tilde{R}(\mathcal{M}_\delta^\ell)$. Since $\tilde{R}(\mathcal{M}_\delta) \geq \tilde{R}(\mathcal{M}_\delta^\ell)$ for all ℓ, we optimize over ℓ to obtain the best possible lower bound on $\tilde{R}(\mathcal{M}_\delta^\ell)$. Recall that $\delta(\ell) \le 1/(4\ell)$.

Theorem 21. [Lower Bound] For any ℓ, we have

$$\tilde{R}(\mathcal{M}_\delta^\ell) \ge 2^{\ell-1}\log n - 2^\ell(\log\frac{1}{\delta(\ell)} + \ell/2) - 2^{\ell-1}(\log 4\pi e\ell + 1),$$

and $\tilde{R}(\mathcal{M}_\delta) \ge \max_\ell \tilde{R}(\mathcal{M}_\delta^\ell)$. □

Before proving this theorem, we consider specific forms for δ to get an idea of the order of magnitude of the redundancy in Theorem 21.

Corollary 22. If $\delta(\ell) = \frac{1}{\ell^c}$ with $c > 1$, then we have

$$\tilde{R}(\mathcal{M}_\delta) = \Omega(n/\log^{2c-1} n),$$

for $\ell = \log n - 2c\log\log n + o(1)$. If $\delta(\ell) = 2^{-c\ell}$, then

$$\tilde{R}(\mathcal{M}_\delta) = \Omega(n^{1/(2c+1)}\log n),$$

for $\ell = \frac{1}{2c+1}\log n + o(1)$. If $\delta(\ell) = 2^{-2^{c\ell}}$, then

$$\tilde{R}(\mathcal{M}_\delta) = \Omega(\log^{1+1/c} n),$$

for $\ell = \frac{1}{c}\log\log n + o(1)$. □

For any $p \in \mathcal{M}_\delta^\ell$, we know that the distribution of p on $\{0,1\}^n$ can be uniquely determined by at most 2^ℓ parameters, i.e. the transition probabilities $p(1 \mid \mathbf{s})$. Let

$$\hat{\mathcal{M}}_\delta^\ell = \{p \in \mathcal{M}_\delta^\ell \mid \forall \mathbf{s} \in S_p,\ |p(1 \mid \mathbf{s}) - 1/2| \le \delta(\ell)\},$$

be a sub-family of $\mathcal{M}_\delta^\ell$ with all the transition probabilities close to 1/2. The following lemma is directly from *Redundancy-Capacity* theorem[43].

Lemma 23. Let $\epsilon_{\mathbf{s}}$ be the maximum mean square error of estimating parameters $p \in \hat{\mathcal{M}}_{\delta}^{\ell}$ from their length n sample. Then

$$\tilde{R}(\hat{\mathcal{M}}_{\delta}^{\ell}) \geq 2^{\ell} \log \delta(\ell) - 2^{\ell-1} \log (2\pi e \epsilon_{\mathbf{s}}) \,. \tag{5.2}$$

Proof

Consider the following Markov chain

$$\hat{\mathcal{M}}_{\delta}^{\ell} \overset{(a)}{\rightarrow} P \overset{(b)}{\rightarrow} X_{-\ell+1}^{n} \overset{(c)}{\rightarrow} \hat{P},$$

where (a) P is a random Markov process chosen from a uniform prior over $\hat{\mathcal{M}}_{\delta}^{\ell}$, (b) X_1^n is a length n sample from distribution P, (c) $\hat{P}$ is an estimate of P from the sample X_1^n that uses the empirical probabilities $\frac{n_{\mathbf{s}1}}{n_{\mathbf{s}}}$ to estimate $P(1 \mid \mathbf{s})$ for any $\mathbf{s} \in \{0,1\}^{\ell}$.

By capacity-redundancy theorem one knows that

$$\tilde{R}(\hat{\mathcal{M}}_{\delta}^{\ell}) \geq I(P; X_1^n) \geq I(P; \hat{P}),$$

where the second inequality is by data processing inequality. Note that

$$\begin{aligned} I(P;\hat{P}) &= h(P) - h(P|\hat{P}) \\ &= h(P) - h(P - \hat{P}|\hat{P}) \\ &\geq h(P) - h(P - \hat{P}), \end{aligned} \tag{5.3}$$

where the last inequality follows since conditioning reduces entropy. To bound first term in (5.3) let $P \in \hat{\mathcal{M}}_{\delta}^{\ell}$ be uniform on the hypercube A with edge lengths $\delta(l)$. Then

$$h(P) = 2^{\ell} \log \delta(\ell).$$

since $h(P) = \log \mathrm{Vol}(A)$.

To bound $h(P-\hat{P})$, let K be the covariance matrix of any estimator of parameter space condition on $\mathbf{x}^n$. Then using Theorem 8.6.5 in [44]

$$h(P-\hat{P}) \leq \frac{1}{2}\log(2\pi e)^{2^\ell}|K|.$$

Let $|K|$ and λ_i show determinant and eigenvalues of matrix K, respectively. Let ϵ_i be the element diagonal elements of covariance matrix. Then by the definition of trace of a matrix

$$\sum_i \epsilon_i = \operatorname{tr}(K) = \sum_i \lambda_i.$$

Using arithmetic-geometric inequality, we get

$$\left(\frac{\sum_i \lambda_i}{2^\ell}\right)^{2^\ell} \geq \prod_i \lambda_i. \tag{5.4}$$

Also $\sum_i \epsilon_i \leq 2^\ell \epsilon_{\mathbf{s}}$. Then

$$|K| = \prod_i \lambda_i \leq \left(\frac{\sum_i \epsilon_i}{2^\ell}\right)^{2^\ell} \leq \epsilon_{\mathbf{s}}^{2^\ell}. \tag{5.5}$$

Applying (5.5) in $\frac{1}{2}\log(2\pi e)^{2^\ell}|K|$, we have

$$h(P-\hat{P}) \leq 2^{\ell-1}\log\big(2\pi e\epsilon_{\mathbf{s}}\big).$$

and lemma follows. □

To bound $\epsilon_{\mathbf{s}}$ one needs to find an estimator that makes it as small as possible. We will show that the empirical estimation $\hat{P}(\mathbf{s}) = \frac{n_{\mathbf{s}1}}{n_{\mathbf{s}}}$, is sufficient to establish our lower bound. We now find an upper bound on the estimation error of state $\mathbf{s}$ using naive estimator.

Lemma 24. Consider the naive estimator $\hat{P}(\mathbf{s}) = \frac{n_{\mathbf{s}1}}{n_{\mathbf{s}}}$. Then,

$$E[(\hat{P}(\mathbf{s}) - P(\mathbf{s}))^2] \leq \min\{E\left[\frac{1}{n_{\mathbf{s}}}\right], 1\}.$$

Proof Note that

$$E[(\hat{P}(\mathbf{s}) - P(\mathbf{s}))^2] = E\big[E[(\hat{P}(\mathbf{s}) - P(\mathbf{s}))^2|n_{\mathbf{s}}]\big].$$

Condition on $n_{\mathbf{s}}$, the symbols follow string $\mathbf{s}$ can be considered as outputs of an *i.i.d.* Bernoulli with parameter $P(\mathbf{s})$. For a sequence of zeros and ones with length n drawn *i.i.d.* from $B(p)$ with k one, it is easy to see that $E[(\frac{k}{n} - p)^2] \leq \frac{1}{n}$, so

$$E\big[E[(\hat{P}(\mathbf{s}) - P(\mathbf{s}))^2|n_{\mathbf{s}}] \leq \min\{E\left[\frac{1}{n_{\mathbf{s}}}\right], 1\}.$$

□

So finding the lower bound on redundancy reduces to find an upper bound on $E[\frac{1}{n_{\mathbf{s}}}]$. We need two following technical lemmas to bound $E[\frac{1}{n_{\mathbf{s}}}]$.

Lemma 25. Let $X_1, X_2, \cdots, X_n$ be binary random variables, such that for any $1 \leq t \leq n$

$$\Pr\left(X_t = 1 \mid X_1, \cdots, X_{t-1}\right) \geq q,$$

for some $q \in [0, 1]$. Then, for any $1 \leq k \leq n$

$$\Pr\left(\sum_{i=1}^{n} X_i \leq k\right) \leq \sum_{i=0}^{k} \binom{n}{i} q^i (1-q)^{n-i}.$$

Proof We use double induction on k and n to prove this theorem.

Consider the base case for $k = 0$, and an arbitrary n, then we need to bound $\Pr\left(\sum_{i=1}^{n} X_i \leq 0\right)$, which is equal to say that $\Pr(X_1 = 0, X_2 = 0, \ldots, X_n = 0)$. But

$$\begin{aligned}\Pr(X_1 = 0, \ldots, X_n = 0) &= \Pr(X_n = 0|X_1 = 0, \ldots, X_{n-1} = 0) \ldots \Pr(X_1 = 0) \\ &\leq (1-q)^n\end{aligned}$$

where the first equation is using chain rule and the inequality follows by the assumption that

$$\Pr\left(X_n = 1 \mid X_1, \cdots, X_{n-1}\right) \geq q.$$

If $k = n$ then

$$\sum_{i=0}^{n} \binom{n}{i} q^i (1-q)^{n-i} = 1$$

so we need to have $\Pr\left(\sum_{i=1}^{n} X_i \leq k\right) < 1$ which holds always.

For the induction step we just need to show that if $(n', k') \leq (n, k)$ and

$$\Pr\left(\sum_{i=1}^{n'} X_i \leq k'\right) \leq \sum_{i=0}^{k'} \binom{n'}{i} q^i (1-q)^{n'-i},$$

holds, then

$$\Pr\left(\sum_{i=1}^{n} X_i \leq k\right) \leq \sum_{i=0}^{k} \binom{n}{i} q^i (1-q)^{n-i}$$

holds. To see it, define

$$A_k^n = \{\sum_{i=1}^{n} X_i \leq k\},$$

$$B_k^n = \{X_1 = 0 \wedge \sum_{i=1}^{n} X_i \leq k\}, \quad \text{and}$$

$$C_k^n = \{X_1 = 1 \wedge \sum_{i=1}^{n} X_i \leq k-1\}.$$

Define

$$T_k^n = \sum_{i=0}^{k} \binom{n}{i} q^i (1-q)^{n-i}.$$

Using chain rule we have

$$\Pr\{B_k^n\} = \Pr\{X_1 = 0\}\Pr\{\sum_{i=1}^{n} X_i \leq k | X_1 = 0\}$$

$$\Pr\{C_k^n\} = \Pr\{X_1 = 1\}\Pr\{\sum_{i=1}^{n} X_i \le k-1 | X_1 = 1\}.$$

Let $P(X_1 = 1) = \tilde{q} > q$, then $P(X_1 = 0) = 1 - \tilde{q} < 1 - q$.

Note that $A_k^n = B_k^n \cup C_k^n$. Using union bound and since B_k^n and C_k^n and are disjoint,

$$\Pr\{A_k^n\} = \Pr\{B_k^n\} + \Pr\{C_k^n\},$$

Then

$$\begin{aligned}
\Pr\{A_k^n\} &= (1-\tilde{q})\Pr\{\sum_{i=1}^{n} X_i \le k | X_1 = 0\} + \tilde{q}\Pr\{\sum_{i=1}^{n} X_i \le k-1 | X_1 = 1\} \\
&\overset{(a)}{\le} (1-q)T_k^{n-1} + qT_{k-1}^{n-1} \\
&= (1-q)\sum_{i=0}^{k} \binom{n-1}{i} q^i (1-q)^{n-1-i} + q\sum_{i=0}^{k-1} \binom{n}{i} q^i (1-q)^{n-i} \\
&= \sum_{i=0}^{k} \binom{n}{i} q^i (1-q)^{n-i}
\end{aligned}$$

and (a) follows by using induction assumption.

□

Lemma 26. For all $p \in \hat{\mathcal{M}}_\delta^\ell$, we have

$$p\left(n_{\mathbf{s}} \le \frac{n}{2\ell 2^\ell} - \sqrt{\frac{n \log n}{\ell 2^\ell}}\right) \le \frac{1}{n},$$

Proof Divide length n sequence to subsequence of length l and let $m = \frac{n}{\ell}$. Let 1_i for $i \in \{1, 2, \ldots, m\}$ as

$$1_i = 1\{\mathbf{s} \prec X_{(i-1)\ell}^{i\ell}\}$$

Note that

$$\begin{aligned}
p(1_i = 1 \mid 1_0, 1_1, \cdots, 1_{i-1}) &\overset{(a)}{\geq} \left(\frac{1}{2} - \delta(\ell)\right)^{\ell} \\
&\geq \frac{1}{2^{\ell}}(1 - 2\delta(\ell))^{\ell} \\
&\overset{(b)}{\geq} \frac{1}{2^{\ell}}(1 - 2\ell\delta(\ell)) \\
&\geq \frac{1}{2^{\ell+1}},
\end{aligned}$$

where (a) follows since $p(1 \mid \mathbf{s}') \in [1/2 - \delta(\ell), 1/2 + \delta(\ell)]$ for all $\mathbf{s}' \in S_p$ and (b) is by union bound. Let $q = \frac{1}{2^{\ell+1}}$ in Lemma 25, then

$$\Pr\big(n_s \leq k\big) \leq \sum_{i=0}^{k} \binom{m}{i} (\frac{1}{2^{\ell+1}})^i (1 - \frac{1}{2^{\ell+1}})^{m-i}.$$

Right hand side in last inequality is the probability that sum of some *i.i.d.* random variables (we denote it by S) drawn from $\mathcal{B}(\frac{1}{2^{l+1}})$ with mean $\mu = \frac{m}{2^{\ell+1}}$ is less than k. Let $k = (1-\gamma)\mu$ where $0 \leq \gamma \leq 1$ is arbitrary. Then

$$\sum_{i=0}^{k} \binom{m}{i} (\frac{1}{2^{\ell+1}})^i (1 - \frac{1}{2^{\ell+1}})^{m-i} = \Pr(S \leq (1-\gamma)\mu).$$

Using Chernoff bound, we get

$$\Pr(S \leq (1-\gamma)\mu) \leq e^{-\frac{\gamma^2\mu}{2}}.$$

Let $\gamma = 2\sqrt{\frac{2^{\ell}\ell\log n}{n}}$, then

$$e^{-\frac{\gamma^2\mu}{2}} = e^{-2\frac{2^{\ell+1}\ell\log n}{n}\frac{n}{2^{\ell+1}\ell}} \leq \frac{1}{n}.$$

So $k = (1-\gamma)\mu = \frac{n}{2^{\ell+1}\ell} - \sqrt{\frac{n\log n}{2^{\ell}\ell}}$ and lemma follows. □

We now combine Lemma 25 and Lemma 26 to bound $E\big[\frac{1}{n_{\mathbf{s}}}\big]$.

Lemma 27. For n large enough,

$$E\left[\frac{1}{n_{\mathbf{s}}}\right] \leq \frac{\ell 2^{\ell+1}}{n}.$$

Proof Let $k = \frac{n}{2^{\ell+1}\ell} - \sqrt{\frac{n \log n}{2^{\ell}\ell}}$, then

$$\begin{aligned} E\left[\frac{1}{n_{\mathbf{s}}}\right] &= \sum_{\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}} \frac{1}{n_{\mathbf{s}}} p\left(\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}\right) + \sum_{\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}} \frac{1}{n_{\mathbf{s}}} p\left(\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}\right) \\ &\leq \sum_{\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}} p\left(\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}\right) + \frac{1}{k} \sum_{\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}} p\left(\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}\right), \end{aligned}$$

where the inequality is by the fact that $\frac{1}{n_{\mathbf{s}}} < 1$. Using Lemma 26

$$\sum_{\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}} p\left(\frac{1}{n_{\mathbf{s}}} \geq \frac{1}{k}\right) < \frac{1}{n}.$$

Also $\sum_{\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}} p\left(\frac{1}{n_{\mathbf{s}}} < \frac{1}{k}\right) < 1$. So

$$E\left[\frac{1}{n_{\mathbf{s}}}\right] \leq \frac{1}{n} + \frac{1}{k}.$$

But $k = \frac{n}{2^{\ell+1}\ell} - \sqrt{\frac{n \log n}{2^{\ell}\ell}} = \frac{n}{2^{\ell+1}\ell}(1 - 2\sqrt{\frac{2^{\ell}\ell \log n}{n}})$, then

$$E\left[\frac{1}{n_{\mathbf{s}}}\right] \leq \frac{1}{n} + \frac{\ell 2^{\ell+1}}{n}\left(\frac{1}{1 - 2\sqrt{\frac{2^{\ell}\ell \log n}{n}}}\right).$$

But we can choose n large enough so that $\sqrt{\frac{2^{\ell}\ell \log n}{n}} < \frac{1}{16}$, then

$$\frac{1}{1 - 2\sqrt{\frac{2^{\ell}\ell \log n}{n}}} < 2.$$

So

$$E\left[\frac{1}{n_{\mathbf{s}}}\right] \leq \frac{1+2\ell 2^{\ell+1}}{n} \simeq \frac{2\ell 2^{\ell+1}}{n}.$$

□

We are now ready to give proof of theorem 21.

Proof of Theorem 21

$$\begin{aligned}
\tilde{R}(\hat{\mathcal{M}}_\delta^\ell) &\overset{(a)}{\geq} 2^\ell \log \delta(\ell) - 2^{\ell-1} \log\left(2\pi e \epsilon_{\mathbf{s}}\right) \\
&\overset{(b)}{\geq} 2^\ell \log \delta(\ell) - 2^{\ell-1} \log\big(2\pi e E[\frac{1}{n_s}]\big) \\
&\overset{(c)}{\geq} 2^\ell \log \delta(\ell) - 2^{\ell-1} \log\big(2\pi e \frac{2\ell 2^{\ell+1}}{n}\big) \\
&= 2^{\ell-1} \log n - 2^{\ell-1} \log 4\pi e \ell \\
&\qquad - 2^{\ell-1} \log 2^{\ell+1} - 2^\ell \log \frac{1}{\delta(\ell)} \\
&= 2^{\ell-1} \log n - 2^\ell (\log \frac{1}{\delta(\ell)} + \ell/2) - 2^{\ell-1}(\log 4\pi e \ell + 1)
\end{aligned}$$

where (a) is using Lemma 23, (b) is by Lemma 24 and (c) follows by Lemma 27.

5.4 The Upper Bound

To obtain the upper bound, we first show that for any given $\mathbf{x} \in \{0,1\}^n$, the maximum probability from any distribution in $\mathcal{M}_\delta$ will not much greater than that from $\mathcal{M}_\delta^\ell$ for an appropriate choice of ℓ. This allows us a simple upper bound based on truncating the memory of sources. Unfortunately (or fortunately), this simple argument does not allow for tight matching bounds—hence we will need to refine our argument further.

Lemma 28. Fix any past $x_{-\infty}^0$. For any $\mathbf{x} \in \{0,1\}^n$, let $\hat{p}(\mathbf{x}) = \max_{p \in \mathcal{M}_\delta} p(\mathbf{x}|x_{-\infty}^0)$ and $\hat{p}_\ell(\mathbf{x}) = \max_{p \in \mathcal{M}_\delta^\ell} p(\mathbf{x}|x_{-\infty}^0)$. Then

$$\hat{p}(\mathbf{x}) \leq 2^{2n\delta(\ell)} \hat{p}_\ell(\mathbf{x}).$$

Proof The continuity condition implies that for any $p \in \mathcal{M}_\delta$, we can find $p_\ell \in \mathcal{M}_\delta^\ell$ such that for all $a \in \{0,1\}$, $\mathbf{w} \in \{0,1\}^\ell$ and $\mathbf{s} \in S_p(\mathbf{w})$, we have

$$p(a \mid \mathbf{s}) \leq (1+\delta(\ell))p_\ell(a \mid \mathbf{w}).$$

Thus we have $p(\mathbf{x}) \leq (1+\delta(\ell))^n p_\ell(\mathbf{x}) \leq 2^{2n\delta(\ell)}\hat{p}_\ell(\mathbf{x})$, where the last inequality follows since $(1+\delta(\ell))^n \approx e^{n\delta(\ell)}$. □

Proposition 5.

$$R(\mathcal{M}_\delta) \leq \min_\ell 2^{\ell-1}\log n + 2n\delta(\ell).$$

Proof Shtarkov's sum [45] gives

$$2^{R(\mathcal{M}_\delta)} = \sum_{\mathbf{x}\in\{0,1\}^n} \hat{p}(\mathbf{x}).$$

Thus by Lemma 28, we have

$$2^{R(\mathcal{M}_\delta)} \leq 2^{2n\delta(\ell)} \sum_{\mathbf{x}\in\{0,1\}^n} \hat{p}_\ell(\mathbf{x}) = 2^{2n\delta(\ell)}2^{R(\mathcal{M}_\delta^\ell)}.$$

Therefore,

$$R(\mathcal{M}_\delta) \leq 2n\delta(\ell) + R(\mathcal{M}_\delta^\ell).$$

Observe that,

$$R(\mathcal{M}_\delta^\ell) \leq R(\mathcal{M}^\ell) \leq 2^{\ell-1}\log n,$$

where the last inequality holds whenever $\ell > 1$, see *e.g.* [6]. □

As before, we work out the above bounds for specific δ.

Corollary 29. If $\delta(\ell) = \frac{1}{\ell^c}$ with $c > 1$, then

$$R(\mathcal{M}_\delta) = O(n/\log^{c-1} n),$$

for $\ell = \log n - c\log\log n + o(1)$. For $\delta(\ell) = 2^{-c\ell}$, we have

$$R(\mathcal{M}_\delta) = O(n^{1/(c+1)} \log n),$$

for $\ell = \frac{1}{c+1}\log n + o(1)$. For $\delta(\ell) = 2^{-2^{c\ell}}$, we have

$$R(\mathcal{M}_\delta) = O(\log^{1+1/c} n),$$

, for $\ell = \frac{1}{c}\log\log n$. □

Comparing Corollary 2 and Corollary 8, the upper and lower bounds on the redundancy have asymptotically tight order when δ diminishes doubly exponentially. For polynomial δ the lower bound and upper bound orders differ by $\log n$ factors. However, for δ to be exponential, we have a polynomial gap between the lower and upper bound.

This suggests that either the lower or upper or both bound are too loose. For the lower bound, recall that the main contribution came from the fast mixing sources in M, while the other sources—the ones that are problematic to estimate, were summarily ignored.

Yet we will show in what follows that our lower bound given in Theorem 1 is actually tight. We need the following technical lemmas to refine our upper bound

Lemma 30. [Extended Azuma inequality] Let $X_1, \cdots, X_k, \cdots$ be martingale differences with $|X_i| \leq 1$, τ is a stopping time (i.e. event $\{\tau = k\}$ only depends on $\sigma(X_1, \cdots, X_k)$). If $\tau \leq n$, then we have

$$\Pr\left(\left|\sum_{i=1}^{\tau} X_i\right| \geq \gamma\sqrt{\tau}\right) \leq ne^{-\gamma^2/2}.$$

Proof Define $A_k = \{|X_1 + \cdots + X_k| \geq \gamma\sqrt{k}\}$, $B_k = \{\tau = k\}$, let $C_k = A_k \cap B_k$. In fact, C_n is the event that we stop at n while it is a wrong time to stop. Note that $|X_i - X_{i-1}| < 1$, using Azuma inequality we have

$$\Pr[A_n] = \Pr\{|\sum_{i=1}^{n} X_i| \geq \gamma\sqrt{n}\} \leq e^{-\gamma^2/2}.$$

Then

$$\Pr[\cup_{k=1}^{n} C_k] \leq \sum_{k=1}^{n} \Pr[A_k \cap B_k] \leq \sum_{k=1}^{n} \Pr[A_k] \leq ne^{-\gamma^2/2}.$$

and the Lemma follows. □

Lemma 31. For any $p \in \mathcal{M}_\delta$, we have

$$p\left(\sum_{\mathbf{w}\in\{0,1\}^\ell} |n_{\mathbf{w}1} - n_{\mathbf{w}}\tilde{p}_{\mathbf{w}}| \leq \log n\sqrt{n2^\ell}\right) \geq 1 - \frac{2^\ell}{n^3},$$

for n large enough that $\log n \geq 6$, where $\tilde{p}_{\mathbf{w}}$ is defined in Section II.

Proof Define

$$1_i(\mathbf{s}) = \begin{cases} 1, & \text{the } i\text{-th appearance of } \mathbf{w} \text{ in } \mathbf{w} \prec \mathbf{s} \\ & 0, \quad \text{otherwise.} \end{cases}$$

Let $W_i = \sum_{w\preceq s} 1_i(\mathbf{s})p(1|\mathbf{s})$, and define

$$Y_i(\mathbf{w}) = \begin{cases} 1, & \text{the } i\text{-th appearance of } \mathbf{w} \text{ happens follows by one} \\ & 0, \quad \text{otherwise.} \end{cases}$$

Let $Z_i = Y_i - W_i$, then by Lemma 2 in [46], Z_i are martingale differences and $|Z_i| < 1$.

Note that $n_{\mathbf{w}1} = \sum_i Y_i$ and $n_{\mathbf{w}}\tilde{p}_{\mathbf{w}} = \sum_i W_i$ and

$$|n_{\mathbf{w}1} - \tilde{p}_{\mathbf{w}}n_{\mathbf{w}}| = \sum_{\mathbf{s}\in S_{\mathbf{w}}(p)} |n_{\mathbf{s}1} - p(1|\mathbf{s})n_{\mathbf{s}}| = |\sum_{i=1}^{n_{\mathbf{w}}} Z_i|$$

Define $z_{\mathbf{w}} = |n_{\mathbf{w}1} - \tilde{p}_{\mathbf{w}}n_{\mathbf{w}}|$. Then using Lemma 30,

$$\begin{aligned} p\bigg(z_{\mathbf{w}} \geq \log n\sqrt{n_{\mathbf{w}}}\bigg) &\leq ne^{-\log n^2/2} \\ &= ne^{\log n^{-\log n/2}} \\ &= \frac{n}{n^{\log n/2}} \\ &\leq \frac{1}{n^3}. \end{aligned}$$

Let $A_{\mathbf{w}} = \{z_{\mathbf{w}} < \log n\sqrt{n_{\mathbf{w}}}\}$. Then

$$\begin{aligned} p(\cup_{\mathbf{w}} A^c_{\mathbf{w}}) &= p(\cup_{\mathbf{w}}\{z_{\mathbf{w}} \geq \log n\sqrt{n_{\mathbf{w}}}\}) \\ &\leq \sum_{\mathbf{w}\in\{0,1\}^\ell} p(z_{\mathbf{w}} \geq \log n\sqrt{n_{\mathbf{w}}}) \\ &\leq \sum_{\mathbf{w}\in\{0,1\}^\ell} \frac{1}{n^3} \\ &= \frac{2^\ell}{n^3}. \end{aligned}$$

Therefore,

$$p(\cap_{\mathbf{w}} A_{\mathbf{w}}) = 1 - p(\cup_{\mathbf{w}} A^c_{\mathbf{w}}) \geq 1 - \frac{2^\ell}{n^3}.$$

Note that event $\{\cap_{\mathbf{w}} A_{\mathbf{w}}\}$ implies event $\{\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}} < \sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}}\}$, so

$$p\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}} < \sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}}\Big) \geq p(\cap_{\mathbf{w}} A_{\mathbf{w}}) \geq 1 - \frac{2^\ell}{n^3}.$$

Also,

$$p\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}} < \sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}}\Big) = p\Big(\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}}\Big)^2 < (\sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}})^2\Big).$$

Using Cauchy-Schwartz inequality we have,

$$\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}}\Big)^2 \leq \sum_{\mathbf{w}\in\{0,1\}^\ell} \log^2 n \sum_{\mathbf{w}\in\{0,1\}^\ell} n_{\mathbf{w}} = n2^\ell \log^2 n.$$

So,

$$p\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}} < \sum_{\mathbf{w}\in\{0,1\}^\ell} \log n\sqrt{n_{\mathbf{w}}}\Big) = p\Big(\sum_{\mathbf{w}\in\{0,1\}^\ell} z_{\mathbf{w}} < \sqrt{n2^\ell}\log n\Big),$$

and the Lemma follows. □

Consider a sample $\mathbf{x}$ from $p \in \mathcal{M}_\delta$, past $x^0_{-\infty}$ and consider the empirical aggregated probabilities in (5.1) for $\mathbf{w} \in \{0,1\}^\ell$. We now consider a memory ℓ Markov source that has its conditional probability of 1 given $\mathbf{w} \in \{0,1\}^{\ell}$ equal to the empirically aggregated probabilities in (5.1), call the source $\tilde{p}_\ell$. Note that $\tilde{p}_\ell$ need not be in the class $\mathcal{M}_\delta^\ell$ or $\mathcal{M}_\delta$, and while we do not explicitly say so in notation for ease of readability, $\tilde{p}_\ell$ depends on the sample $\mathbf{x}$.

For any $\mathbf{x}$ and $p \in \mathcal{M}_\delta$, and for $\mathbf{w} \in \{0,1\}^\ell$, let $z_\ell = \sum_{\mathbf{w}} z_{\mathbf{w}}$.

Lemma 32. For any $p \in \mathcal{M}_\delta$ and $\mathbf{x} \in \{0,1\}^n$, we have

$$\tilde{p}_{\ell+1}(\mathbf{x}) \le 2^{2n\delta^2(\ell)+2z_{\ell+1}\delta(\ell)}\tilde{p}_\ell(\mathbf{x}).$$

Moreover, we have

$$p\left(\left\{\mathbf{x} : z_{\ell+1} \le \log n\sqrt{n2^{\ell+1}}\right\}\right) \ge 1 - \frac{2^\ell}{n^3}$$

Proof

Note that

$$\tilde{p}_{\ell+1}(\mathbf{x}) = \prod_{\mathbf{w}\in\{0,1\}^l} \tilde{p}_{1\mathbf{w}}^{n_{1\mathbf{w}1}}(1-\tilde{p}_{1\mathbf{w}})^{n_{1\mathbf{w}}-n_{1\mathbf{w}1}}\tilde{p}_{0\mathbf{w}}^{n_{0\mathbf{w}1}}(1-\tilde{p}_{0\mathbf{w}})^{n_{0\mathbf{w}}-n_{0\mathbf{w}1}}$$

, and

$$\tilde{p}_\ell(\mathbf{x}) = \prod_{\mathbf{w}\in\{0,1\}^\ell} \tilde{p}_{\mathbf{w}}^{n_{\mathbf{w}1}}(1-\tilde{p}_{\mathbf{w}})^{n_{\mathbf{w}}-n_{\mathbf{w}1}}$$

So we just need to show that

$$\prod_{\mathbf{w}\in\{0,1\}^l} \tilde{p}_{1\mathbf{w}}^{n_{1\mathbf{w}1}}(1-)^{n_{1\mathbf{w}}-n_{1\mathbf{w}1}}\tilde{p}_{0\mathbf{w}}^{n_{0\mathbf{w}1}}(1-\tilde{p}_{0\mathbf{w}})^{n_{0\mathbf{w}}-n_{0\mathbf{w}1}}$$
$$\le 2^{2n\delta^2(\ell)+2z_{\ell+1}\delta(\ell)} \prod_{\mathbf{w}\in\{0,1\}^\ell} \tilde{p}_{\mathbf{w}}^{n_{\mathbf{w}1}}(1-\tilde{p}_{\mathbf{w}})^{n_{\mathbf{w}}-n_{\mathbf{w}1}}$$

To see it, note

$$\tilde{p}_{\mathbf{w}}n_{\mathbf{w}} = \tilde{p}_{1\mathbf{w}}n_{1\mathbf{w}} + \tilde{p}_{0\mathbf{w}}n_{0\mathbf{w}}.$$

Let

$$\tilde{p}_{1\mathbf{w}} = \tilde{p}_{\mathbf{w}} + \tilde{p}_{\mathbf{w}}\delta_1,$$
$$\tilde{p}_{0\mathbf{w}} = \tilde{p}_{\mathbf{w}} + \tilde{p}_{\mathbf{w}}\delta_0,$$

for some $|\delta_1| < l$ and $|\delta_0| < l$. Then

$$n_{0\mathbf{w}}\delta_0 + n_{1\mathbf{w}}\delta_1 = 0.$$

Let

$$n_{1\mathbf{w}1} = \tilde{p}_{1\mathbf{w}}n_{1\mathbf{w}} + z_1' = \tilde{p}_{\mathbf{w}}n_{1w} + \tilde{p}_w n_{1w}\delta_1 + z_1',$$
$$n_{0\mathbf{w}1} = \tilde{p}_{0\mathbf{w}}n_{0\mathbf{w}} + z_0' = \tilde{p}_{\mathbf{w}}n_{0\mathbf{w}} + \tilde{p}_{\mathbf{w}}n_{0\mathbf{w}}\delta_0 + z_0',$$

for some z_0' and z_1'. Also

$$\begin{aligned}
\log \tilde{p}_{1\mathbf{w}}^{n_{1\mathbf{w}1}}(1-\tilde{p}_{1\mathbf{w}})^{n_{1\mathbf{w}}-n_{1\mathbf{w}1}} &= n_{1\mathbf{w}1}\log\tilde{p}_{\mathbf{w}} \\
&\quad + n_{1\mathbf{w}0}\log(1-\tilde{p}_{1\mathbf{w}}) \\
&\leq n_{1\mathbf{w}1}(\log\tilde{p}_{1\mathbf{w}} + \delta_1) \\
&\quad + n_{1\mathbf{w}0}(\log(1-\tilde{p}_{\mathbf{w}}) - \frac{\tilde{p}_{\mathbf{w}}}{1-\tilde{p}_{\mathbf{w}}}\delta_1)) \\
&= A_{1\mathbf{w}} + (\tilde{p}_{\mathbf{w}} + \frac{\tilde{p}_{\mathbf{w}}^2}{1-\tilde{p}_{\mathbf{w}}})n_{1\mathbf{w}}\delta_1^2 \\
&\quad + (\frac{\tilde{p}_{\mathbf{w}}}{1-\tilde{p}_{\mathbf{w}}})\delta_1 z_1' \\
&\leq A_{1\mathbf{w}} + 2n_{1\mathbf{w}}\delta(\ell)^2 + 2\delta(\ell)|z_1'|.
\end{aligned}$$

where

$$A_{1\mathbf{w}} = n_{1\mathbf{w}1}\log\tilde{p}_{\mathbf{w}} + n_{1\mathbf{w}0}\log(1-\tilde{p}_{\mathbf{w}}).$$

Similarly,

$$\begin{aligned}\log \tilde{p}_{0\mathbf{w}}^{n_{0\mathbf{w}1}}(1-\tilde{p}_{0\mathbf{w}})^{n_{0\mathbf{w}}-n_{0\mathbf{w}1}} &= n_{0\mathbf{w}1}\log \tilde{p}_{0\mathbf{w}} + n_{0\mathbf{w}0}(1-\log \tilde{p}_{0\mathbf{w}})\\ &\le n_{0\mathbf{w}1}(\log \tilde{p}_{\mathbf{w}} + \delta_0)\\ &\quad + n_{0\mathbf{w}0}(\log(1-\tilde{p}_{\mathbf{w}}) - \frac{\tilde{p}_{\mathbf{w}}}{1-\tilde{p}_{\mathbf{w}}}\delta_0))\\ &= A_{0\mathbf{w}} + (\tilde{p}_{\mathbf{w}} + \frac{\tilde{p}_{\mathbf{w}}^2}{1-\tilde{p}_{\mathbf{w}}})n_{0w}\delta_0^2\\ &\quad + (\frac{\tilde{p}_{\mathbf{w}}}{1-\tilde{p}_{\mathbf{w}}})\delta_0 z_0'\\ &\le A_{0\mathbf{w}} + 2n_{0\mathbf{w}}\delta(l)^2 + 2\delta(l)|z_0'|,\end{aligned}$$

and $A_{0\mathbf{w}} = n_{0\mathbf{w}1}\log \tilde{p}_{\mathbf{w}} + n_{0\mathbf{w}0}\log(1-\tilde{p}_{\mathbf{w}})$. Summing over all $\mathbf{w}$, we have

$$\begin{aligned}&\sum_{\mathbf{w}} \log \tilde{p}_{1\mathbf{w}}^{n_{1\mathbf{w}1}}(1-\tilde{p}_{1\mathbf{w}})^{n_{1\mathbf{w}}-n_{1\mathbf{w}1}}\tilde{p}_{0\mathbf{w}}^{n_{0\mathbf{w}1}}(1-\tilde{p}_{0\mathbf{w}})^{n_{0\mathbf{w}}-n_{0\mathbf{w}1}}\\ &\le \sum_{\mathbf{w}}(A_{1\mathbf{w}} + A_{0\mathbf{w}}) + 2n\delta(\ell)^2 + 2\delta(\ell)z_{\ell+1}\\ &= \log \tilde{p}_{\mathbf{w}}^{n_{\mathbf{w}1}}(1-\tilde{p}_{\mathbf{w}})^{n_{\mathbf{w}}-n_{\mathbf{w}1}} + 2n\delta(\ell)^2 + 2\delta(\ell)z_{\ell+1},\end{aligned}$$

where we use the fact that $z_{\ell+1} = \sum_{\mathbf{w}}|z_0'| + |z_1'|$. Also using Lemma 31 one can see that

$$p\left(\left\{\mathbf{x} : z_{\ell+1} \le \log n\sqrt{n2^{\ell+1}}\right\}\right) \ge 1 - \frac{2^\ell}{n^3}$$

□

Lemma 33. For any $p \in \mathcal{M}_\delta$, we have

$$p\left(\{\mathbf{x} : p(\mathbf{x}) \le 2^{r_\ell}\tilde{p}_\ell(\mathbf{x})\}\right) \ge 1 - \frac{\sum_{k=\ell}^{2\ell} 2^k}{n^3},$$

where

$$r_\ell = n\delta(2\ell) + \sum_{k=\ell}^{2\ell} 2n\delta^2(k) + 2\log n\sqrt{n2^{k+1}}\delta(k).$$

Proof Using Lemma 28 we have

$$p(\mathbf{x}) \leq \tilde{p}_{2\ell}(x) 2^{2n\delta(2\ell)}. \tag{5.6}$$

Also,

$$\begin{aligned}
\tilde{p}_{\ell+1}(\mathbf{x}) &= \prod_{\mathbf{w}\in\{0,1\}^{\ell}} \tilde{p}_{1\mathbf{w}}^{n_{1\mathbf{w}1}}(1-\tilde{p}_{1\mathbf{w}})^{n_{1\mathbf{w}}-n_{1\mathbf{w}1}} \tilde{p}_{0\mathbf{w}}^{n_{0\mathbf{w}1}}(1-\tilde{p}_{0\mathbf{w}})^{n_{0\mathbf{w}}-n_{0\mathbf{w}1}} \\
&\leq 2^{2n\delta^2(\ell)+2z_\ell\delta(\ell)} \prod_{\mathbf{w}\in\{0,1\}^{\ell}} \tilde{p}_{\mathbf{w}}^{n_{\mathbf{w}1}}(1-\tilde{p}_{\mathbf{w}})^{n_{\mathbf{w}}-n_{\mathbf{w}1}}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\tilde{p}_{2\ell+1}(\mathbf{x}) &\leq \sum_{k=\ell}^{2\ell} 2^{2n\delta^2(k)+2z_{k+1}\delta(k)} \prod_{\mathbf{w}\in\{0,1\}^{k}} \tilde{p}_{\mathbf{w}}^{n_{\mathbf{w}1}}(1-\tilde{p}_{\mathbf{w}})^{n_{\mathbf{w}}-n_{\mathbf{w}1}} \\
&= 2^{\sum_{k=\ell}^{2\ell} 2n\delta^2(k)+2z_{k+1}\delta(k)} \tilde{p}_{\ell}(\mathbf{x}). \qquad (5.7)
\end{aligned}$$

and using equation (5.6) and equation (5.7), we have

$$p(\mathbf{x}) \leq \tilde{p}_{\ell}(\mathbf{x}) 2^{n\delta(2\ell)+\sum_{k=\ell}^{2\ell} 2n\delta^2(k)+2z_{k+1}\delta(k)}.$$

but note that for all k

$$p\left(\left\{\mathbf{x} : z_{k+1} \leq \log n\sqrt{n2^{k+1}}\right\}\right) \geq 1 - \frac{2^k}{n^3},$$

Using union bound

$$p\left(\left\{\mathbf{x} : \sum_{k=l}^{2l} z_{k+1} \leq \sum_{k=l}^{2l} \log n\sqrt{n2^{k+1}}\right\}\right) \geq 1 - \sum_{k=l}^{2l} \frac{2^k}{n^3},$$

and the lemma follows.

□

Theorem 34. [Improved Upper Bound] Redundancy of $\mathcal{M}_\delta$ is upper bounded by

$$\tilde{R}(\mathcal{M}_\delta) \leq 2^{\ell-1}\log n + n\delta(2\ell) + \sum_{k=\ell}^{2\ell}\left(n\delta^2(k) + \log n\sqrt{n2^k}\delta(k)\right) + (2^{2\ell+1} - 2^\ell)\frac{n}{n^3}$$

for any integer $\ell \in \mathbb{N}$.

Proof Let $\mathcal{T}_p = \{\mathbf{x} : p(\mathbf{x}) \leq 2^{r_\ell}\tilde{p}_\ell(\mathbf{x})\}$ be the set of good sequences and $\mathcal{T}_p^c = \{\mathbf{x} : \mathbf{x} \notin \mathcal{T}\}$. Let $c(\mathbf{x})$ be the best code for memory ℓ sources. Let $|c(\mathbf{x})|$ denote the length of $c(\mathbf{x})$. Let $q(\mathbf{x}) = \frac{2^{-c(\mathbf{x})}+2^{-n}}{2}$. We can choose $c(\mathbf{x})$ tight enough so that $\sum 2^{-c(\mathbf{x})} = 1$.

Then

$$\begin{aligned}
\tilde{R}(\mathcal{M}_\delta) &= \max_{p\in\mathcal{M}_\delta}\sum_{\mathbf{x}} p(\mathbf{x})\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\\
&= \max_{p\in\mathcal{M}_\delta}\sum_{\mathbf{x}\in\mathcal{T}_p} p(\mathbf{x})\log\frac{p(\mathbf{x})}{q(\mathbf{x})} + \sum_{\mathbf{x}\in\bar{\mathcal{T}}_p} p(\mathbf{x})\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\\
&\leq \max_{p\in\mathcal{M}_\delta}\sum_{\mathbf{x}\in\mathcal{T}_p} p(\mathbf{x})\max_{\mathbf{x}\in\{0,1\}^n}\log\frac{p(\mathbf{x})}{q(\mathbf{x})} + \sum_{\mathbf{x}\in\bar{\mathcal{T}}_p} p(\mathbf{x})\max_{\mathbf{x}\in\{0,1\}^n}\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\\
&\leq \max_{p\in\mathcal{M}_\delta}\sum_{\mathbf{x}\in\mathcal{T}_p} p(\mathbf{x})\max_{\mathbf{x}\in\{0,1\}^n}\log\frac{p_\ell 2^{r_\ell}}{q(\mathbf{x})} + \sum_{\mathbf{x}\in\bar{\mathcal{T}}_p} p(\mathbf{x}).n\\
&\leq \max_{p\in\mathcal{M}_\delta}\max_{\mathbf{x}\in\{0,1\}^n}\log\frac{p_\ell 2^{r_\ell}}{q(\mathbf{x})} + n\frac{\sum_{k=\ell}^{2\ell}2^k}{n^3}\\
&= \max_{p\in\mathcal{M}_\delta}\max_{\mathbf{x}\in\{0,1\}^n}[\log p_\ell(\mathbf{x}) + c(\mathbf{x}) + 1] + r_\ell + n\frac{\sum_{k=\ell}^{2\ell}2^k}{n^3}\\
&= 2^{\ell-1}\log n + r_\ell + (2^{2\ell+1} - 2^\ell)\frac{n}{n^3}.
\end{aligned}$$

Where $r_\ell = n\delta(2\ell) + \sum_{k=\ell}^{2\ell} n\delta^2(k) + \log n\sqrt{n2^\ell}\delta(k)$. Note that first term in the last equation follows since the worst case redundancy of Markov sources with memory ℓ and is bounded by $2^{\ell-1}\log n$. So

$$\begin{aligned}\tilde{R}(\mathcal{M}_\delta) &\le 2^{\ell-1}\log n + n\delta(2\ell)\\ &\qquad + \sum_{k=\ell}^{2\ell}\big(n\delta^2(k) + \log n\sqrt{n2^k}\delta(k)\big) + (2^{2\ell+1}-2^\ell)\frac{n}{n^3}.\end{aligned}$$

□

Corollary 35. For $\delta(\ell) = 2^{-c\ell}$, we have

$$\tilde{R}(\mathcal{M}_\delta) = O(n^{1/(2c+1)}\log n).$$ □

Proof Note that,

$$\begin{aligned}\tilde{R}(\mathcal{M}_\delta) &\le 2^{\ell-1}\log n + n\delta(2\ell)\\ &\qquad + \sum_{k=\ell}^{2\ell}\big(n\delta^2(k) + \log n\sqrt{n2^k}\delta(k)\big) + (2^{2\ell+1}-2^\ell)\frac{n}{n^3},\end{aligned}$$

let $\delta(k) = 2^{-ck}$ then

$$\begin{aligned}\sum_{k=\ell}^{2\ell}\delta^2(k) = \sum_{k=\ell}^{2\ell}2^{-2ck} &= 2^{-2cl} + 2^{-2c(l+1)} + \cdots + 2^{-2c(2\ell)}\\ &= 2^{-2c\ell}\big(1 + 2^{-2c} + \cdots + 2^{-2c\ell}\big)\\ &= 2^{-2c\ell}\frac{1-2^{-2c(\ell+1)}}{1-2^{-2c}}\\ &= \frac{2^{-2c\ell} - 2^{-2c(2\ell+1)}}{1-2^{-2c}},\end{aligned}$$

and

$$\begin{aligned}
\sum_{k=\ell}^{2\ell} 2^{k/2}\delta(k) &= \sum_{k=\ell}^{2\ell} 2^{(-c+1/2)k} \\
&= 2^{(-c+1/2)\ell} + 2^{(-c+1/2)(\ell+1)} + \cdots + 2^{(-c+1/2)2\ell} \\
&= 2^{(-c+1/2)\ell}\big(1 + 2^{-c} + \cdots + 2^{-2c\ell}\big) \\
&= 2^{(-c+1/2)\ell}\frac{1-2^{-c(\ell+1)}}{1-2^{-c}} \\
&= \frac{2^{(-c+1/2)\ell} - 2^{(-2c+1/2)\ell-c}}{1-2^{-c}}.
\end{aligned}$$

Let $\ell = c' \log n$, then

$$\begin{aligned}
\tilde{R}(\mathcal{M}_\delta) \leq \frac{n^{c'}}{2}\log n &+ \frac{n}{n^{-2cc'}} \\
&+ n\left(\frac{n^{-2cc'} - \frac{n^{-4cc'}}{2^{-2c}}}{1-2^{-2c}}\right) \\
&+ \log n\sqrt{n}\left(\frac{n^{c'(-c+1/2)} - \frac{n^{(-2c+1/2)c'}}{2^c}}{1-2^{-c}}\right) \\
&+ (2n^{2c'} - n^{c'})\frac{n}{n^3}.
\end{aligned}$$

Let $c' = \frac{1}{2c+1}$

$$\begin{aligned}
\tilde{R}(\mathcal{M}_\delta) \leq \frac{1}{2}n^{\frac{1}{2c+1}}\log n &+ n^{\frac{1}{2c+1}} \\
&+ \frac{n^{\frac{1}{2c+1}} - \frac{n^{\frac{-2c}{2c+1}}}{2^{-2c}}}{1-2^{-2c}} \\
&+ \log n\left(\frac{n^{\frac{1}{2c+1}} - \frac{1}{2^c}n^{\frac{-c+1}{(2c+1)}}}{1-2^{-c}}\right) \\
&+ (2n^{\frac{2c}{2c+1}} - n^{\frac{1}{2c+1}})\frac{n}{n^3} \\
= \mathcal{O}(n^{\frac{1}{2c+1}}\log n).
\end{aligned}$$

□

Part II

Lossy Compression of Memoryless Sources

6

An Online Polynomial-Time Algorithm for Lossy Compression of Unknown Memoryless Sources

6.1 Introduction

There are two different stories in communication, source coding which is removing redundancy to reduce the number of bits needed to send the data and channel coding which is adding redundancy to have more reliable communication. Source coding also known as data compression can be done with or without distortion. While in some applications it is required to losslessly reconstruct the data, in many of them some amount of distortion is allowed to improve compression rate.

6.1.1 Lempel Ziv Algorithm

In lossless case, an algorithm have been developed by Huffman which is optimal when the underlying distribution of the source is known.

When statistics of the source is unknown, Lempel and Ziv proposed an optimal adaptive polynomial time algorithm which does not need to know the distribution of the data. LZ algorithm has two main parts: distinct parsing and coding.

Distinct Parsing

To explain distinct parsing, let proceed with an example. Consider sequence

$$X = AAABBABABBB.$$

After distinct parsing, every phrase must be the "shortest identical phrase". For example, first parsed phrase in X is "A", the second one is "AA" (It cannot be "A" because "A" already exists in the set of parsed phrases), next one is "B", continuing in a similar way, the set of the parsed phrase of X is

$$\{A, AA, B, BA, BAB, BB\}.$$

Coding

In this step, an index must be assigned to every phrase in the parsed set. Table 6.1 shows the corresponding number of each phrase in the previous example.

Table 6.1: Assigning Index to Parsed Phrases

Phrase	Index
A	0
AA	1
B	2
BA	3
BAB	4
BB	5

To code each phrase it is enough to send an index which points to the part of the phrase that we have already seen. For example, the code assigned to "BAB" is $(4, B)$ where 4 refers to the index of the phrase "BA" and B is the last letter in "BAB". Table 2 lists the assigned codes for each sequence.

Table 6.2: Coding Parsed Phrase

Phrase	Index	Code
A	1	(0,A)
AA	2	(1,A)
B	3	(0,B)
BA	4	(3,A)
BAB	5	(4,B)
BB	6	(3,B)

6.1.2 Length of The Code

The maximum number of bits needed to compress a sequence with LZ algorithm is $(\log_2 c(n) + \log_2 \alpha)$ where $c(n)$ is the number of parsed phrases and α is the alphabet size. In fact, $\log_2 c(n)$ and $\log_2 \alpha$ refer to first (number) and the second part (alphabet) of the codes, respectively. Therefore, total number of the bits needed to code a sequence is $c(n)\big(\log_2 c(n) + \log_2 \alpha\big)$.

6.2 Lossy Compression

There are many applications that some amount of distortion is tolerable to reduce the number of bits required to code. This alternative compression problem is known as lossy compression and has been formulated by Shannon [47] which is also known as Rate-Distortion problem.

6.2.1 Prior Work

Unlike lossless compression, there is no “optimal sequential universal adaptive” algorithm for lossy case. We briefly review some of the works in this area with emphasis on the LZ extension algorithms.

One of the early work for extending LZ lossless to the lossy version is in [48]. Morita and Kobayashi presented a lossy version of the LZW algorithm. Between multiple matches they choose the one with minimum distortion. Also, a fixed-database version for LZ lossy is in [49] and [50], but Yang and Kieffer [51] showed that all these fixed-database extensions of the Lempel-Ziv algorithm are suboptimal. Later, Zamir and Rose [52] used a natural type selection method.

Attallah, et.al [53] and Navarro [54] used an approximate match method and Yang and Kieffer [55] have proposed an exponential-time universal Lempel-Ziv-type block codes. To see more about exponential time algorithms look at [56].

In a non-LZ approach, Jalalli and Weissman [57] used a Markov Chain Monte Carlo method. Korada and Urbanke [58] applied polar codes for source coding. In a very recent work, Jun Muramatsu [59] used a constraint random number generator to present a lossy source coding algorithm.

6.2.2 Rate Distortion

The rate distortion problem has been formulated by Shannon [47] where he showed that *"rate distortion function"* is a lower bound for the compression rate. To define rate distortion function we introduce some notation. Let $X^n = X_1X_2 \dots X_n$ be a sequence drawn *i.i.d.* according to $p(X = x)$, $x \in \mathcal{X}$. Let $Y^n = Y_1Y_2 \dots Y_n$ be the lossy representation of X^n, $Y \in \hat{\mathcal{X}}$. Let $d(X^n, Y^n)$ be the distortion measure between X^n and Y^n where

$$d(X^n, Y^n) = \sum_{i=1}^{n} d(X_i, Y_i)$$

and $d(x, y)$ is the Hamming distortion given by

$$d(x, y) = \begin{cases} 1, & \text{if} \quad x \neq y \\ 0, & \text{if} \quad x = y. \end{cases}$$

Let $E[d(X^n, Y^n)]$ be the expected distortion. The rate distortion function is defined as

$$r(D) \triangleq \min_{p(y|x):E[d(X^n,Y^n)] \leq D} I(X;Y)$$

where $I(X;Y)$ is the mutual information between X and Y [44] and the minimization is over all possible conditional distributions $p(y|x)$ which satisfies $E[d(X^n, Y^n)] < D$. The rate-distortion theorem states that the $r(D)$ is the asymptotic lower bound for compression of the sequence constraint to distortion D.

6.2.3 Optimal Reproduction Type

Let be $p^*(y|x)$ be the optimal distribution achieves rate-distortion function. The optimal reproduction distribution is

$$q^*(y) = \sum_{x \in \mathcal{X}} p(x)p^*(y|x)$$

For a sequence X^n, type of the sequence, $\tau(X^n)$, is defined as total number of one in the sequence over length of the sequence. For example, for $X^9 = 111010111$, $\tau(X^9) = 0.7778$. A sequence Y^n has *"optimal reproduction type"* if it generates using optimal reproduction distribution.

6.3 Codelet Parsing

Codelet parsing is a lossy Lempel-Ziv type algorithm first presented in [60]. In the case of no distortion it reduces to lossless LZ algorithm. It sequentially parses a sequence to phrases which we call "sourcelet" and maps them to "codelet" in the dictionary. We use an example to better explain the algorithm. Let $X^{11} = 10111001101$ and $d = 0.5$. We initialize the dictionary with $\{0, 1\}$ (Figure 6.1). Therefore, in the beginning, the codelets in the dictionary are $\{0\}$ and $\{1\}$. We parse 10111001101 to get a sourcelet which is in distortion 0.5 of a codelet in the dictionary. In the first step, the only possible sourcelet is $\{1\}$. The distortion between sourcelet $\{1\}$ and codelet $\{0\}$ is 1 which is greater than 0.5 and the distortion between sourcelet $\{1\}$ and codelet $\{1\}$ is 0 which is less than 0.5, so the only possible codelet is $\{1\}$ and consequently the chosen codelet is $\{1\}$. Now, we extend the chosen codelet $\{1\}$ in the dictionary to $\{11, 10\}$ (Figure 6.2) and the unparsed string is 0111001101. In this step, possible sourcelets are $\{0, 01\}$. Although sourcelet $\{01\}$ is in distortion 0.5 of codelet $\{11\}$, we only consider *"strong match"* which means that not only $\{11\}$ must be in distortion 0.5 of sourcelet $\{01\}$, but also all of their prefix must satisfy distortion constraint. (We will explain strong match in the next section in more detail). In this case, length one prefix of $\{01\}$,(i.e. 0) has higher distortion than 0.5 with a length one prefix of $\{11\}$. So it can not be a candidate. Therefore, we choose sourcelet $\{0\}$ and codelet $\{0\}$. Now, we extend the codelet $\{0\}$ in the dictionary (Figure 6.3) and the unparsed string will be 111001101. In the next step, the only possible sourcelet is $\{11\}$ and the codelets satisfying the distortion level are $\{01, 11, 10\}$. However, the codelets which strongly match with $\{11\}$ are $\{11, 10\}$. (Later we describe in detail how we choose in the case of multiple strong matches.). The

Table 6.3: An example of evolution of Codelet parsing

i	String	Dictionary	PS	PC	CC
1	10111001101	$\{0,1\}$	$\{1\}$	$\{1\}$	$\{1\}$
2	0111001101	$\{0,10,11\}$	$\{0,01\}$	$\{0,11\}$	$\{0\}$
3	111001101	$\{01,00,10,11\}$	$\{11\}$	$\{01,11,10\}$	$\{11\}$
4	1001101	$\{01,00,10,110,111\}$	$\{10,100\}$	$\{10,00,100\}$	$\{10\}$
5	01101	$\{01,00,110,111,101,100\}$	$\{01,011\}$	$\{01,00,111\}$	$\{01\}$
6	101	$\{00,110,111,101,100,010,011\}$	$\{10,101\}$	$\{00,101,100,111\}$	$\{101\}$

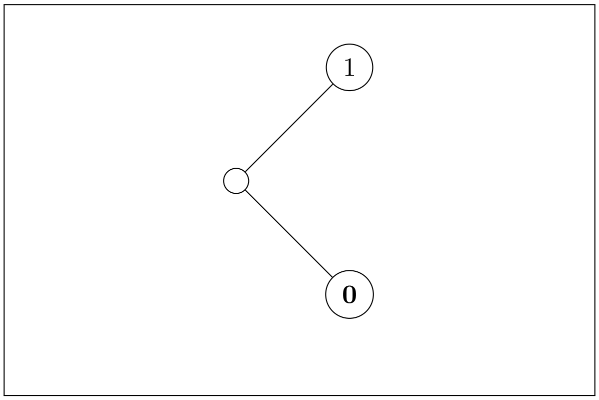


Figure 6.1: String: 10111001101 Possible Sourcelets: $\{1\}$, Possible Codelets: $\{1\}$,Chosen Codelet: $\{1\}$

algorithm choose codelet $\{11\}$ and the dictionary updates as Figure 6.4. Table 6.3 shows the Codelet parsing algorithm applied to string 10111001101 for $d = 0.5$. Abbreviations PS, PC, and CC stand for "Possible Sourcelets", "Possible Codelets", and "Chosen Codelet", respectively. Figures 6.1 to 6.6 show the evolution of the dictionary in each iteration.

6.3.1 Multiple Matches

Here, we propose a mathematical definition for "strong match" and justify the motivation for our definition. This part is the main difference of the algorithm with its previous version in [61], [62] and [63]. An initial version of the current algorithm was proposed in [11]. In the classical LZ algorithm which considers exact matching, we need to traverse just one branch. Figure 6.7a, shows the traversed path for finding exact match of $s = 1101$. In approximate

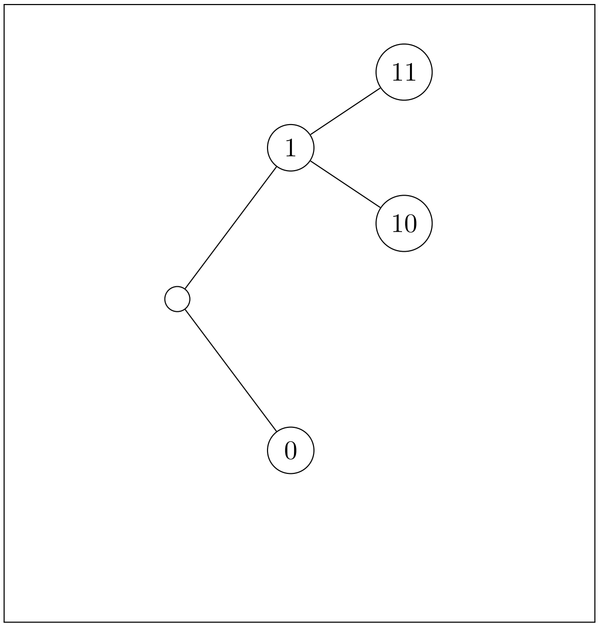


Figure 6.2: String: 0111001101, Possible Sourcelets: $\{0, 01\}$, Possible Codelets: $\{0, 11\}$, Chosen Codelet: $\{0\}$

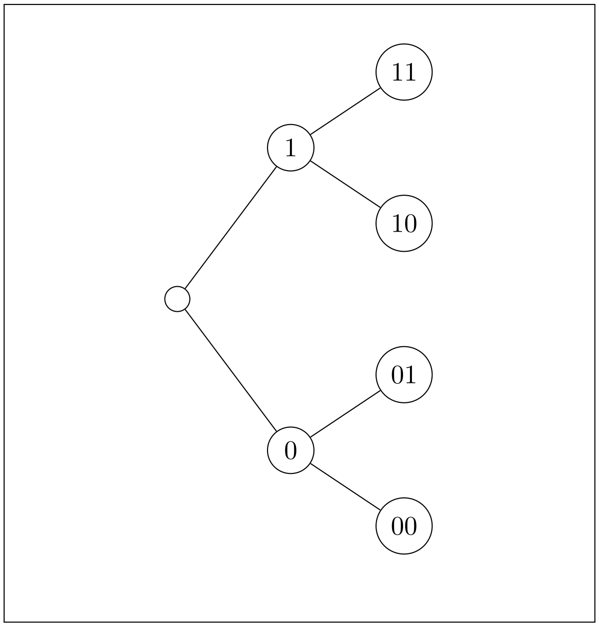


Figure 6.3: String: 111001101, Possible Sourcelets: $\{11\}$, Possible Codelets: $\{01, 11, 10\}$, Chosen Codelet: $\{11\}$

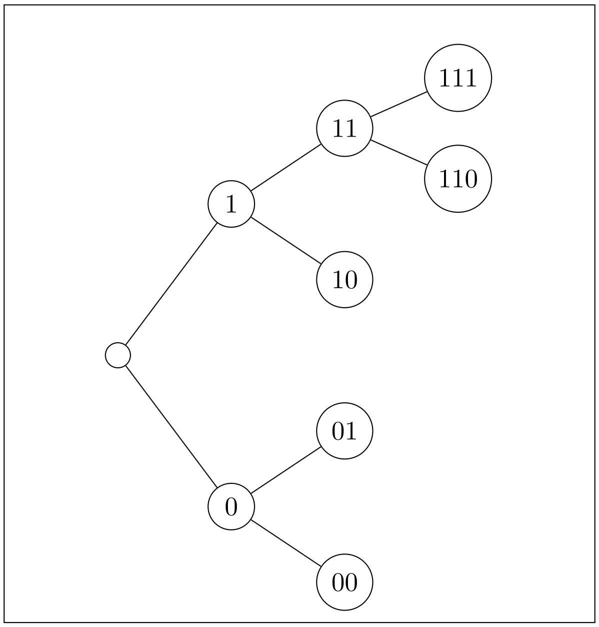


Figure 6.4: String: 1001101, Possible Sourcelets: $\{10, 100\}$, Possible Codelets: $\{10, 00, 100\}$, Chosen Codelet: $\{10\}$

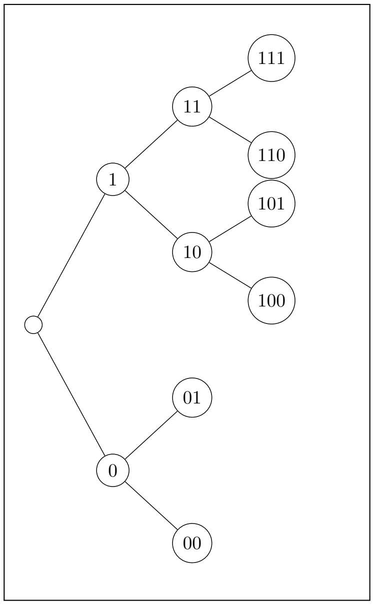


Figure 6.5: String: 01101, Possible Sourcelets: $\{01, 011\}$, Possible Codelets: $\{01, 00, 111\}$, Chosen Codelet: $\{01\}$

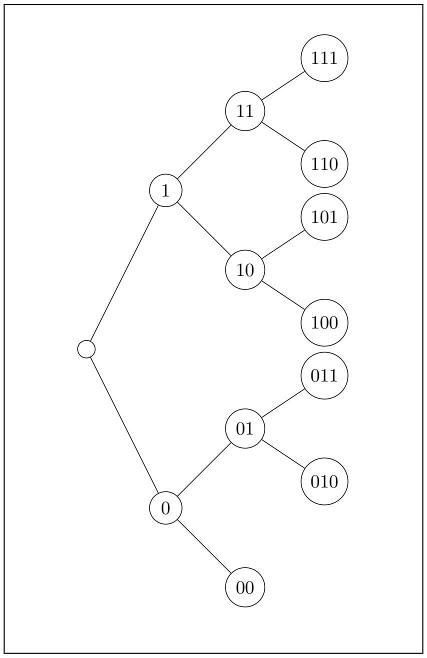


Figure 6.6: String: 101, Possible Sourcelets: $\{10, 101\}$, Possible Codelets: $\{00, 101, 100, 111\}$, Chosen Codelet: $\{101\}$

matching, we must search all branches to find all possible matches so we need to traverse multiple branches. Figure 6.7b shows the possible matches for string $s = 1101$. But how can we reduce number of possible branches without losing "good" matches?

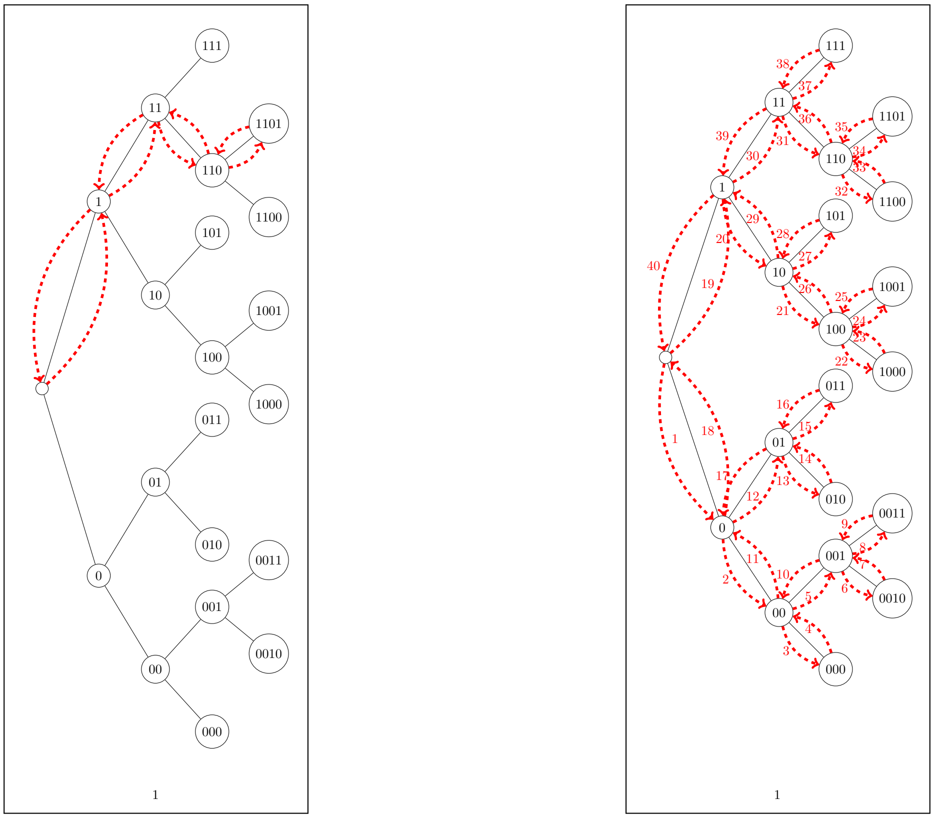
We define concept "strong match" to explain the next step of the algorithm. For a given distortion d,

Definition 4. Two sequences $x^n, \hat{x}^n$ *match* if $1/n \sum_{i=1}^{n} d_H(x_i, \hat{x}_i) < d, d \leq 0.5$. □

Definition 5. Two sequences $x^n, \hat{x}^n$ ***strongly*** *match* if $1/j \sum_{i=1}^{j} d_H(x_i, \hat{x}_i) < d, 1 \leq j \leq n, d \leq 0.5$. □

Assume in an specific epoch of algorithm, string is 1101 and dictionary is as Figure 6.7b. We use strong match to reduce the complexity. Figure 6.8 shows how the strong match reduces the possible matches. But how do we know that using strong match does not omit the sequence obtained from optimal reproduction distribution from the dictionary?

The cycle lemma assures us that even with strong match there are enough possible matches for a given string. In fact, using strong matches helps us to have a smart search.



(a) Exact match for $s = 1101$

(b) possible approximate match for $s = 1101$

Figure 6.7: Exact match vs approximate match

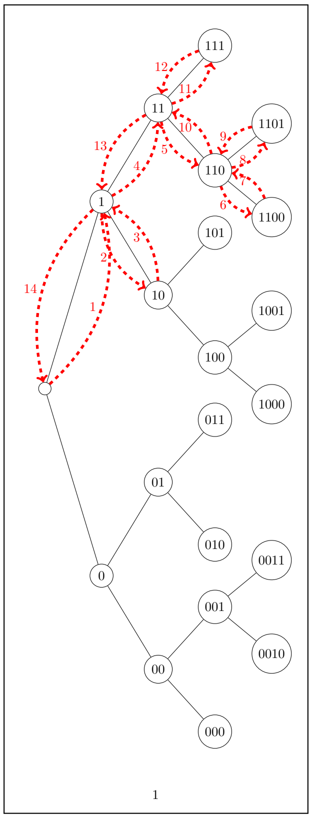


Figure 6.8: Strong match reduces possible branches

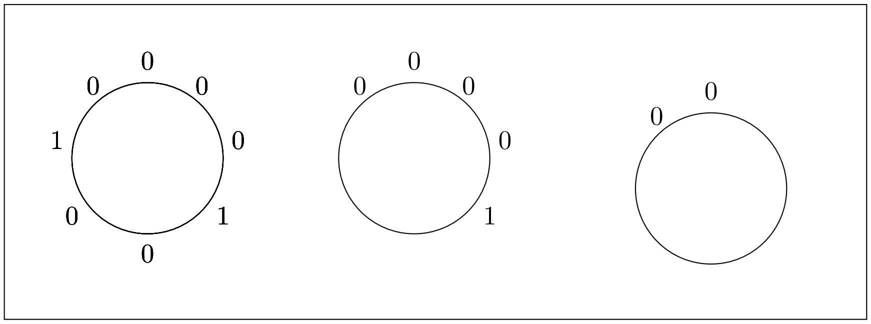


Figure 6.9: Proof of cycle lemma, $X = 00001001$, $k = 2$

6.3.2 Cycle Lemma

A sequence $p_1p_2...p_{m+n}$ of zeros and ones is k-dominating if number of zeros in every subsequence $p_1p_2...p_i$, $1 \leq i \leq m+n$ is greater than k times of number ones. For example, sequence "00010010" is 2-dominating, "00101001" is 1-dominating and "10000000" and "00110001" are not even 1-dominating.

Lemma 36. [64] Let X be any sequence containing m zeros and n ones where $m \geq kn$. Then number of cyclic permutation of X which are $k-$dominating is $m - kn$.

Proof Write sequence X on a circle. Removing a subsequence containing k "zeros" followed by a "one" has no net effect on the result. By removing all such sequences there is $m - kn$ zeros which can be a start point of a permutation. As long as $m \geq kn$, by pigeon-hole principle, there exists such subsequence (Figure 6.9). □

6.4 Variation of the Codelet Parsing Algorithm

Choosing a codelet affects the way we form the dictionary (the codebook), thus selecting between multiple candidates is a delicate task. Not only we need to find a codelet which reconstructs the corresponding sourcelet and satisfy the distortion constraint, but also this codelet will add to dictionary and its leaves will be the next possible matches. But even using strong match cons1traint there can be 1multiple matches for 1a certain sourcelet. There are different ways to choose between them in this case.[1] Perhaps the easiest way is choosing a candidate that has been selected first.Another naive way is choosing the longest candidate. We implemented both methods. Simulation results show that in non of these methods type of the leaves of the dictionary goes to the optimal reproduction type. But in an optimal dictionary, type of the leaves converges to the "optimal reproduction type" asymptotically. In fact, in each step we may choose a candidate which its type has minimum distance to "optimal reproduction type". For example, for $\mathcal{B}(p)$, $q^* = \frac{p-d}{1-2d}$. But in general the underlying distribution is unknown and we don't know the q^*. We simulate the algorithm assuming that we know q^* and from now on we refer to it as "optimal reproduction type" method.[2]

In this section we present simulation results using three different approaches: (i) "Optimal Reproduction Type" method, (ii) "First Match" method, and (iii) "Longest Match" method which correspond to choosing a codelet "closest to optimal reproduction type", a codelet which "has joined to tree sooner" and a codelet which "has the longest length", respectively. By "closest to optimal reproduction type", we simply mean that the absolute value of the difference between type of the chosen codelet and optimal reproduction type is the minimum among all codelets that strongly match with the sourcelet.

6.4.1 Optimal Reproduction Type

6.4.1.1 Compression Rate

Figures 6.10 to 6.12 show the compression rate of the codelet parsing algorithm versus $\log n$ for different p, d and n using "optimal reproduction type" method.

The plots for codelet rate reveals a linear relationship between compression rate and block length n. We use Matlab curve fitting to find the parameters of a simple linear curve for

[1] Note that this step is after adding the strong match constraint. There are a few other papers discussing choosing between possible candidates but their search space is much bigger than here.

[2] We resolve this problem in the next chapter by developing a method that learn the "optimal reproduction type"

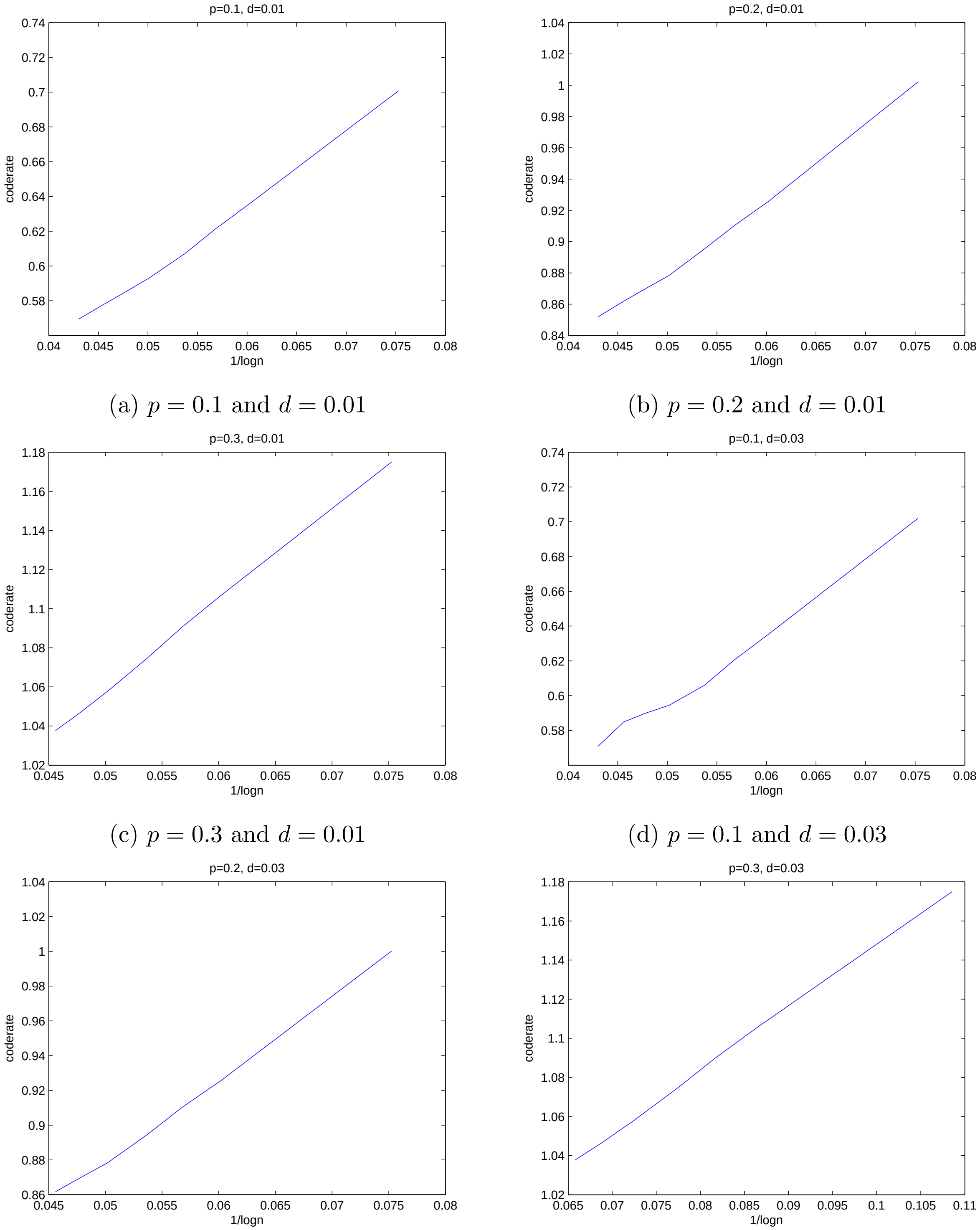
Table 6.4: Curve fitting parameters for compression rate of “Optimal Reproduction Type” method

p	d	Optiaml Rate	a	b	Norms of Residuals
0.1	0.01	0.3882	4.1052	0.38938	0.0059518
0.2	0.01	0.6411	4.6803	0.64638	0.0075957
0.3	0.01	0.8005	4.6661	0.82459	0.0022734
0.1	0.03	0.2746	3.9942	0.39698	0.010753
0.2	0.03	0.5275	4.7179	0.64329	0.0050521
0.3	0.03	0.6869	3.2343	0.82459	0.0022734
0.1	0.05	0.1826	4.2348	0.24596	0.025721
0.3	0.05	0.5949	3.3878	0.81001	0.0061762
0.5	0.1	0.5310	1.1876	1.0413	0.0082403
0.375	0.1	0.4854	3.4434	0.8215	0.0075243
0.375	0.2	0.2325	6.3382	0.41135	0.0041889
0.375	0.3	0.0731	6.8354	0.12459	0.0024781
0.25	0.05	0.5249	3.6944	0.70688	0.0026598
0.25	0.1	0.3423	4.9278	0.52882	0.0059882
0.25	0.2	0.0894	6.6757	0.074766	0.0012464

different p and d. In fact if we approximate compression rate r by $r = a * (1/\log n) + b$, table 6.4 below gives the corresponding parameters a and b.

6.4.1.2 Running Time

Since we are interested in an implementable algorithm, it is very important to monitor how the processing time increases as length of the sequence increases. Figures 6.13 and 6.14 show the running time t vs logarithm of block length n for “Optimal Reproduction Type” method. We use curve fitting to fit a linear model to each figure. The linear model estimates a and b in equation $\log t = a \log n + b$, where from the curve fitting parameters, a is around 1.7 to 2 and b is in range -17 to -21. It shows that the time complexity is of order $\mathcal{O}(n^2)$.



(a) $p = 0.1$ and $d = 0.01$

(b) $p = 0.2$ and $d = 0.01$

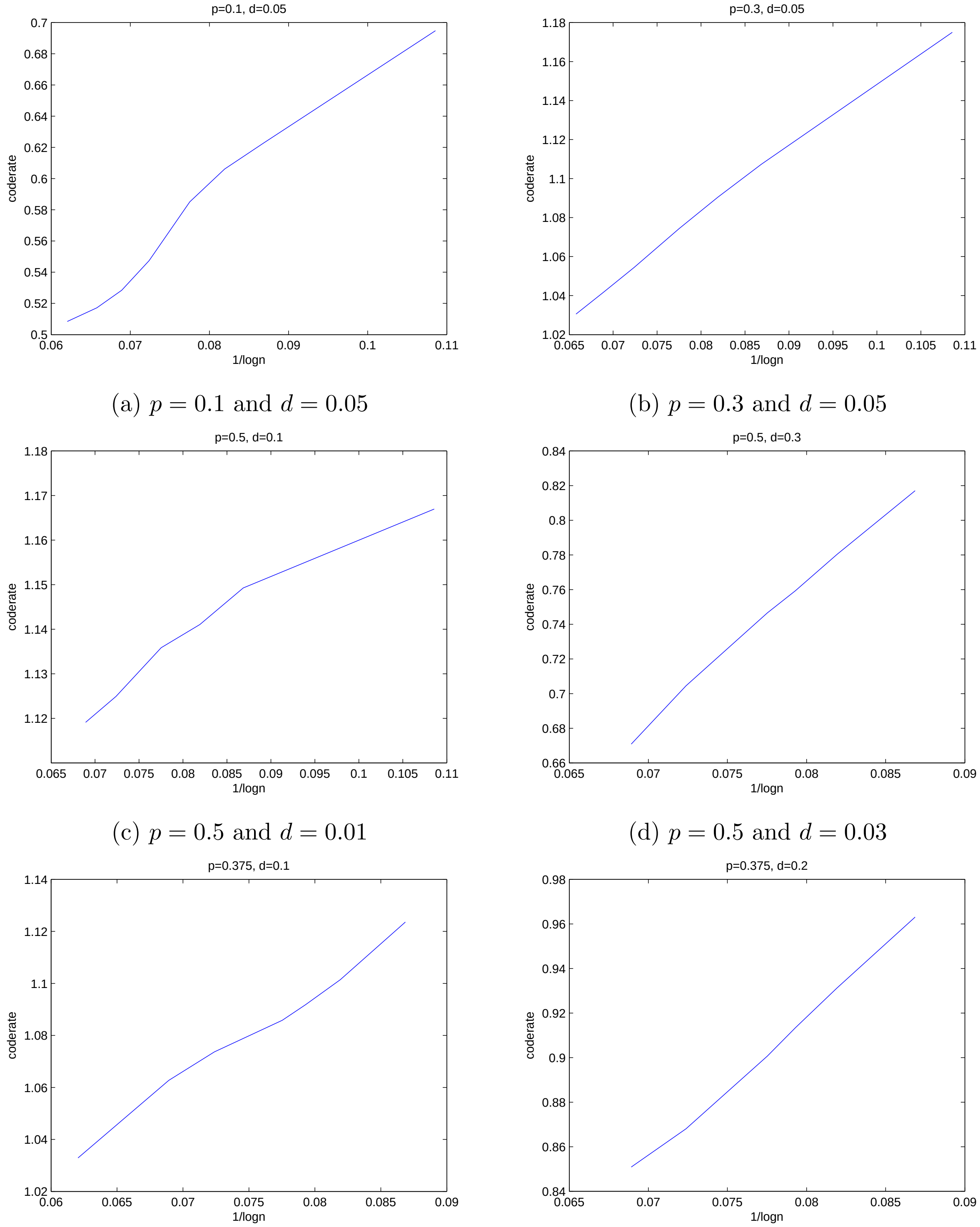
(c) $p = 0.3$ and $d = 0.01$

(d) $p = 0.1$ and $d = 0.03$

(e) $p = 0.2$ and $d = 0.03$

(f) $p = 0.3$ and $d = 0.03$

Figure 6.10: Compression rate using "Optimal Reproduction Type" method



(a) $p = 0.1$ and $d = 0.05$

(b) $p = 0.3$ and $d = 0.05$

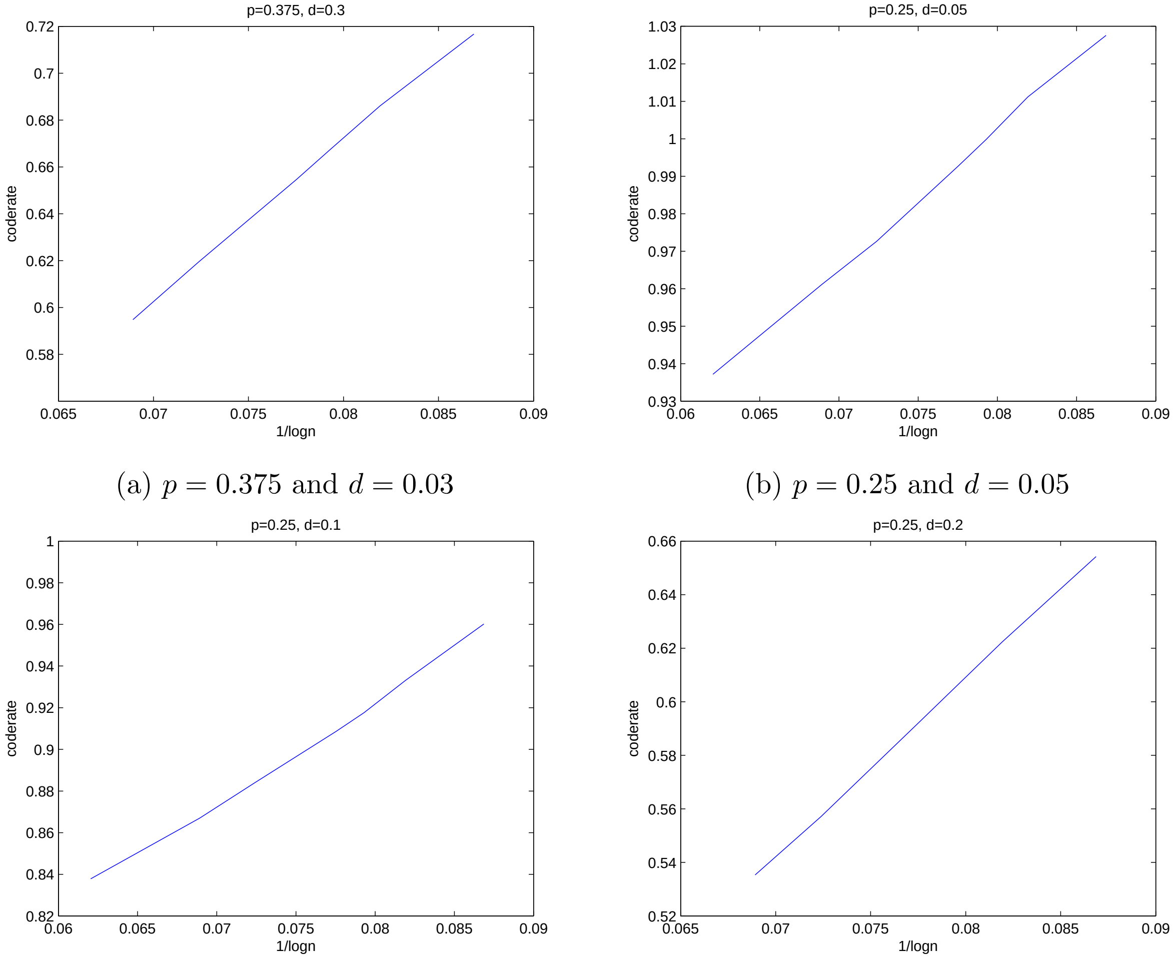
(c) $p = 0.5$ and $d = 0.01$

(d) $p = 0.5$ and $d = 0.03$

(e) $p = 0.375$ and $d = 0.01$

(f) $p = 0.375$ and $d = 0.02$

Figure 6.11: Compression rate using “Optimal Reproduction Type” method



(a) $p = 0.375$ and $d = 0.03$

(b) $p = 0.25$ and $d = 0.05$

(c) $p = 0.25$ and $d = 0.1$

(d) $p = 0.25$ and $d = 0.2$

Figure 6.12: Compression rate using "Optimal Reproduction Type" method

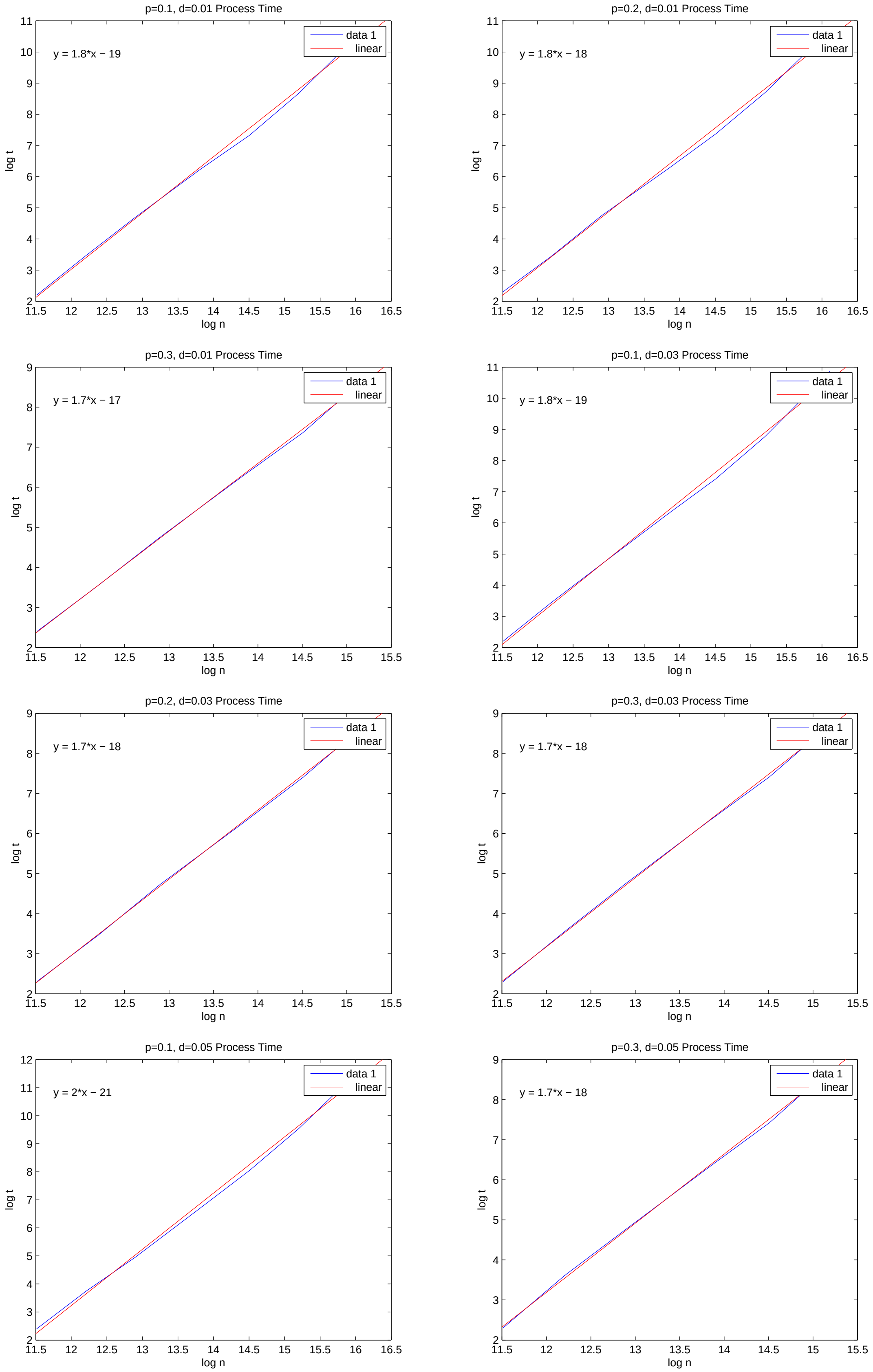


Figure 6.13: Time Complexity for different values of p and d

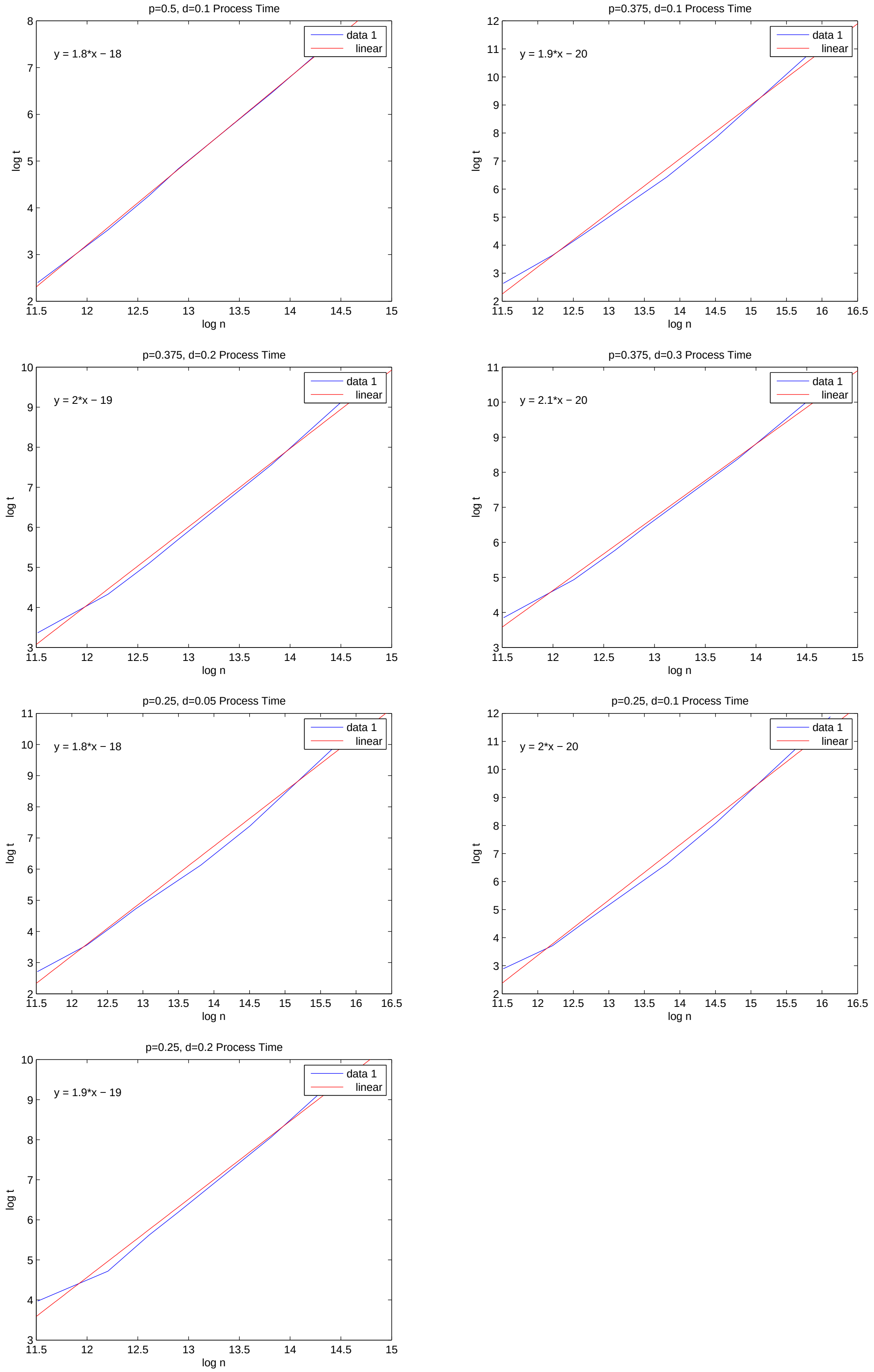


Figure 6.14: Time Complexity for different values of p and d

Table 6.5: “Optimal Reproduction Type” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
10000	0.1	0.01	674	0	0.3882	0.7007
100000	0.1	0.01	4804	9	0.3882	0.6356
200000	0.1	0.01	8809	32	0.3882	0.6212
400000	0.1	0.01	16207	109	0.3882	0.6071
1000000	0.1	0.01	36708	510	0.3882	0.5933

Table 6.6: “Longest Match” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
10000	0.1	0.01	674	0	0.3882	0.700731
100000	0.1	0.01	4804	9	0.3882	0.63557
200000	0.1	0.01	8809	32	0.3882	0.621244
400000	0.1	0.01	16207	109	0.3882	0.607128
1000000	0.1	0.01	36708	534	0.3882	0.593341

6.4.2 “First Match” and “Longest Match”

As we explained before, in the case of multiple strong match, we can simply choose one with longest lengths or the first match. We plot the compression rate for different p, d and n. In the case of small distortion, the results are very similar to “Optimal Reproduction Type” method (Tables 6.5 to 6.7 show this similarity). But for other values “optimal reproduction type” apparently works better.

Tables 6.8 to 6.10 compare the compression rate and time complexity of these methods.

Table 6.7: “First Match” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
10000	0.1	0.01	674	0	0.3882	0.7007
100000	0.1	0.01	4804	19	0.3882	0.63557
200000	0.1	0.01	8809	31	0.3882	0.621244
400000	0.1	0.01	16207	109	0.3882	0.607128
1000000	0.1	0.01	36708	521	0.3882	0.593341

Table 6.8: “Optimal Reproduction Type” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
100000	0.25	0.2	4931	53	0.0894	0.654229
200000	0.25	0.2	8823	112	0.0894	0.622333
400000	0.25	0.2	15843	489	0.0894	0.592194
1000000	0.25	0.2	34646	3189	0.0894	0.557122

Table 6.9: “Longest Match” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
100000	0.25	0.2	6040	20	0.0894	0.819044
200000	0.25	0.2	11021	62	0.0894	0.795053
400000	0.25	0.2	20208	236	0.0894	0.773089
1000000	0.25	0.2	45434	1491	0.0894	0.748365

Table 6.10: “First Match” method

n	p	d	NumLeaf	ProcTime	targRate	ComRate
100000	0.25	0.2	8233	58	0.0894	1.15321
200000	0.25	0.2	15309	250	0.0894	1.14068
300000	0.25	0.2	22063	593	0.0894	1.13473
400000	0.25	0.2	28583	1127	0.0894	1.12923
1000000	0.25	0.2	65716	8094	0.0894	1.11743

6.4.3 Analysis of Type

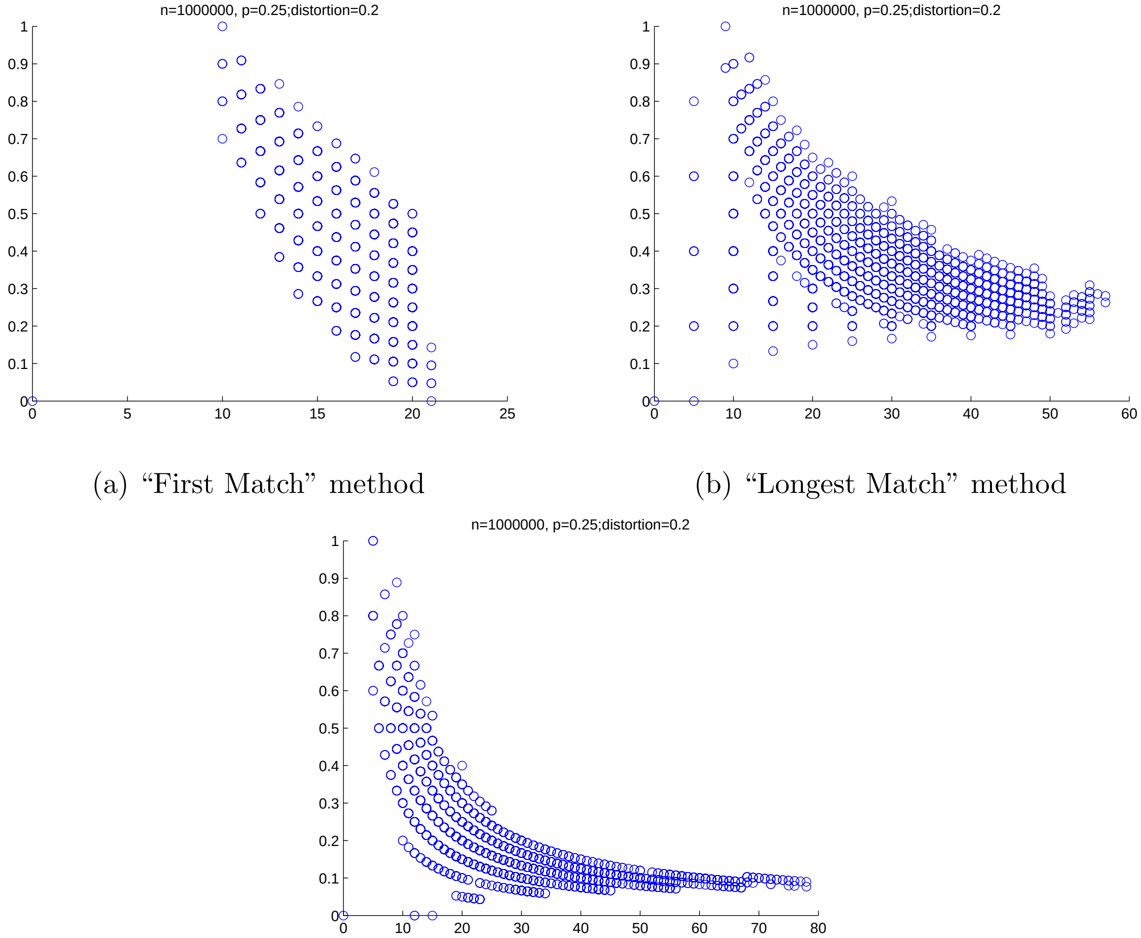
In an optimal set up, type of the leaves of the dictionary converges to "Optimal Reproduction Type", so monitoring them helps us to get a better insight of the way the algorithm evolves. Usually (but not necessarily) longer leaves come in the later steps of the algorithm. We plot type of all 6f the leaves of the dictionary versus their length. (Since different leaves may have same length and type, a singlet dot in plot may represent more than one leave. We later use heat plots to show if more than on leave have a specific length and type.) Different methods result in different type evolution. We plot type of the leaves versus length of the leaves for different method for some p, d, and n. Figures 6.15 refers to $n = 10^6$, $p = 0.25$ and $d = 0.2$, so $q^* = 0.083$ and Figure 6.16 refers to $n = 10^6$, $p = 0.1$ and $d = 0.05$, so $q^* = 0.056$. Simulation results show that using "Optimal Reproduction Type" method, type of the leaves of the tree are more close to q^* compare to "First Match" and "Longest Match". This is analogous to the results we obtained so far that compression rate of "Optimal Reproduction Type" method is better than other variants of the algorithm.

6.5 Learning the Optimal Reproduction Type

In the previous section, we showed that when we use the optimal reproduction type, q^*, to select between multiple matches, the compression rate is better than the case when we choose either the first match or the longest match. However, since the underlying distribution is unknown we do not know what is the optimal reproduction type.

6.5.1 Algorithm

To keep the algorithm universal, we propose a method that learns the optimal reproduction type. In this method, we count the number of times a leaf is a possible candidate (and may or may not be selected) and call it *"match index"*. When we have more than one candidate,

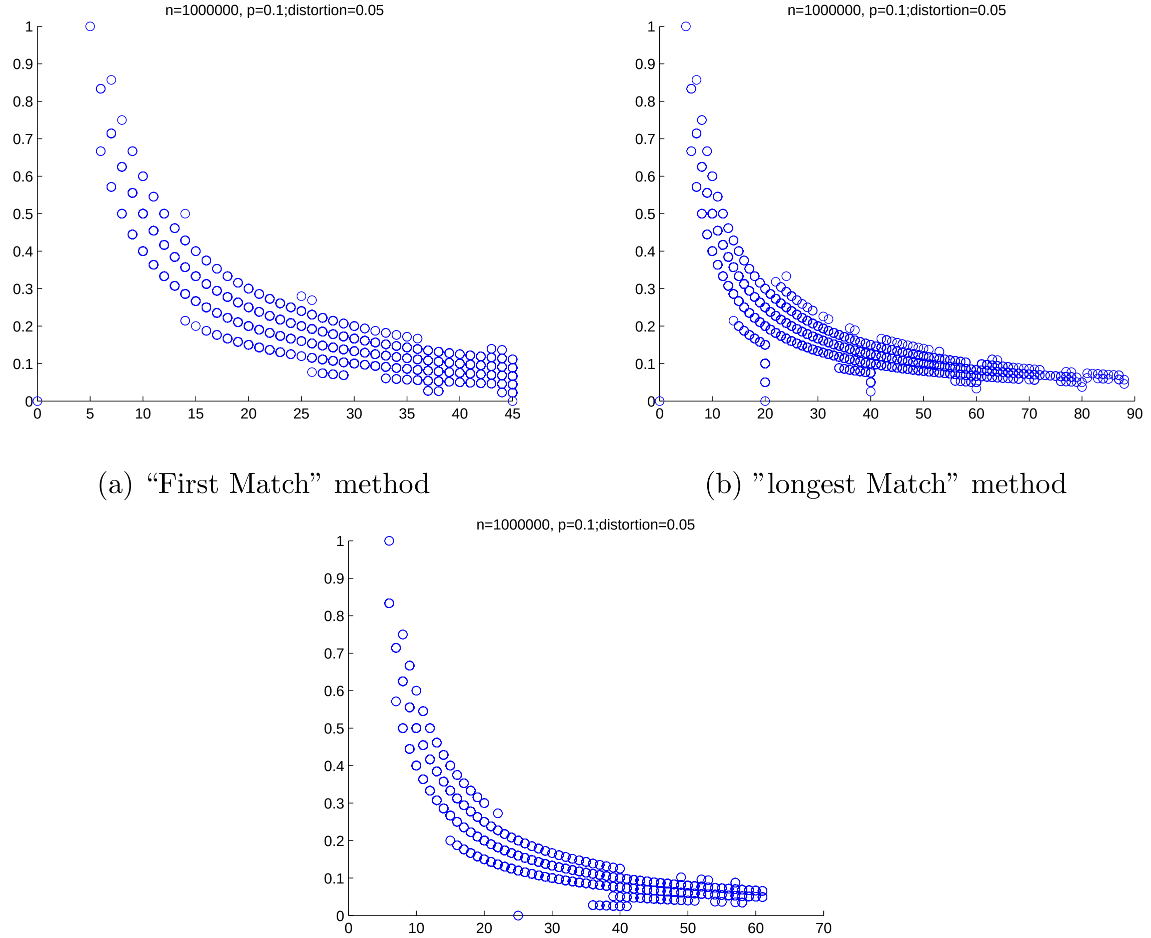


(a) “First Match” method

(b) “Longest Match” method

(c) “Optimal Reproduction Type” method

Figure 6.15: Type of the leaves vs length of the leaves for $n = 10^6$, $p = 0.25$ and $d = 0.2$



(a) “First Match” method

(b) ”longest Match” method

(c) “Optimal Reproduction Type” method

Figure 6.16: Type of the leaves vs length of the leaves for $n = 10^6$, $p = 0.1$ and $d = 0.05$

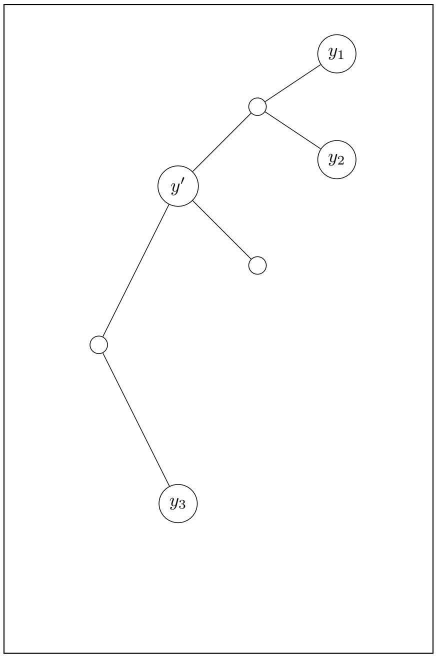


Figure 6.17: Comparing three leaves at different depth

we compare all candidates in an appropriate level. For example, let y_1, y_2 and y_3 be the three leaves which match with a given phrase (Figure 6.17). Let y_3 be the deepest leaf (i.e. it has the shortest length). Then to compare their *"match index"*, the algorithm reaches the parent of y_1 and y_2 at the same depth that y_3 is (which is y') and compare its "match index" (which has been updated based on the number of matches of its leaves). If the parents have the same *"match index"* there are two possibilities: We can either (a) choose one of them at random, or (b) keep both leaves of the parents and then compare their *"match index"* in the next level. The variant (b) works better than (a) and even better than the case we know the optimal reproduction type for some n, p and d.

6.5.2 Simulation Results

Table 6.11 is the simulation results for different p, d and n. Comparing with "First match" and "Longest Match" method, the results show that this method works obviously better than them and its performance is close to knowing the true q^*.

Table 6.11: Performance of the “Most Match” method, “nLeafMost” shows the number of the leaves in the dictionary, and “TimeMost” is the processing time of the algorithm

n	p	d	nLeafMost	TimeMost	compressionRate
100000	0.1	0.01	4804	11	0.63557
100000	0.1	0.03	4778	15	0.631756
100000	0.2	0.03	6747	12	0.92569
100000	0.25	0.1	6826	40	0.937676
100000	0.2	0.05	6731	13	0.923265
100000	0.25	0.2	4852	1574	0.642617
1000000	0.1	0.01	36708	513	0.593341
1000000	0.1	0.03	35859	830	0.578407
1000000	0.2	0.03	52647	536	0.878366
1000000	0.25	0.1	50242	2899	0.834852
1000000	0.2	0.05	51519	679	0.857936

6.5.2.1 Type plots

The type plot we used in section 6.4.3 can be misleading since it may indicate that type of the leaves converges to zero, however the number of leaves with type zero are very small. So instead of a simple type plot we use a heat plot where the color indicates the number of leaves with specific type. Figure 6.18 shows the heat plots for different set of p, n, and d.

6.5.2.2 Codelet Frequency

Figures 6.19 to 6.21 shows leaves with specific length vs their length” and “log of number of leaves with specific length vs their length”.

6.5.2.3 Type of Reconstructed version

To get a better insight of how the leaves of the dictionary evolve, we plot type of the leaves as we parse the sequence. Figure 6.22 and 6.23 show $|q^* - q_i|$ versus i where q_i is the type of the reconstructed sequence y till iteration i. More specifically, there are i leaves in the dictionary at iteration i. It is interesting that for higher distortion, $|q^* - q_i|$ is still decreasing

Figure 6.18: Heat plot for most match method

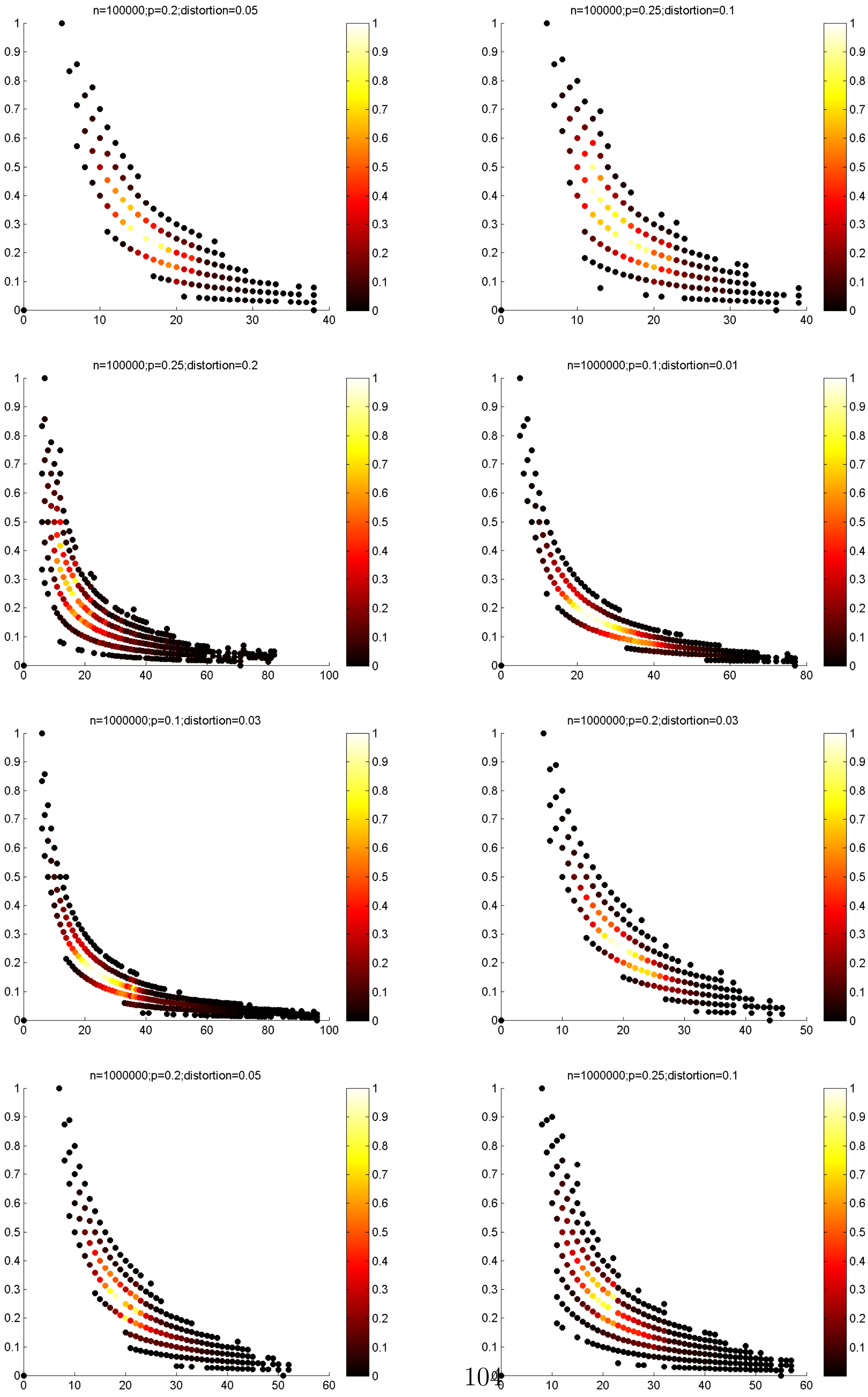
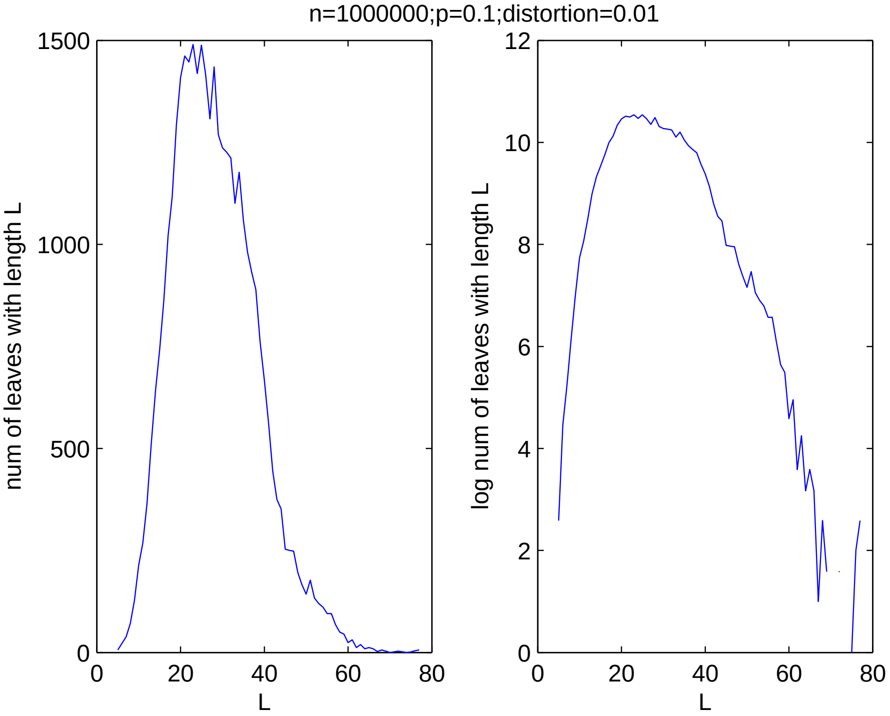


Figure 6.19: Codelet Frequency



which shows that convergence time for higher distortion is longer, but then there is a hope that their compression rate improves by increasing n.

6.5.2.4 Achieved distortion vs desired distortion

In all variants of the Codelet parsing algorithm, real distortion is much less than than the desired distortion, because we ask every phrase to satisfy the distortion level. For example, consider parsed sequence $X = 1, 0, 11, 01, 111, 00, 011, 1111, 000, 110$ and let the distortion level be 0.25. If we just consider a possible strong match for sourcelet 1111, two codelets are possible: 1111 and 1110, but considering a prefix of length 17, all 16 possible phrases of length 4 are a valid candidate. To see how the real distortion is different than the reconstructed distortion we have simulated the algorithm for different n, p, and d. Table 6.12 shows the simulation results. The last column shows the number of positions that X^n and Y^n are different.

Figure 6.20: Codelet Frequency

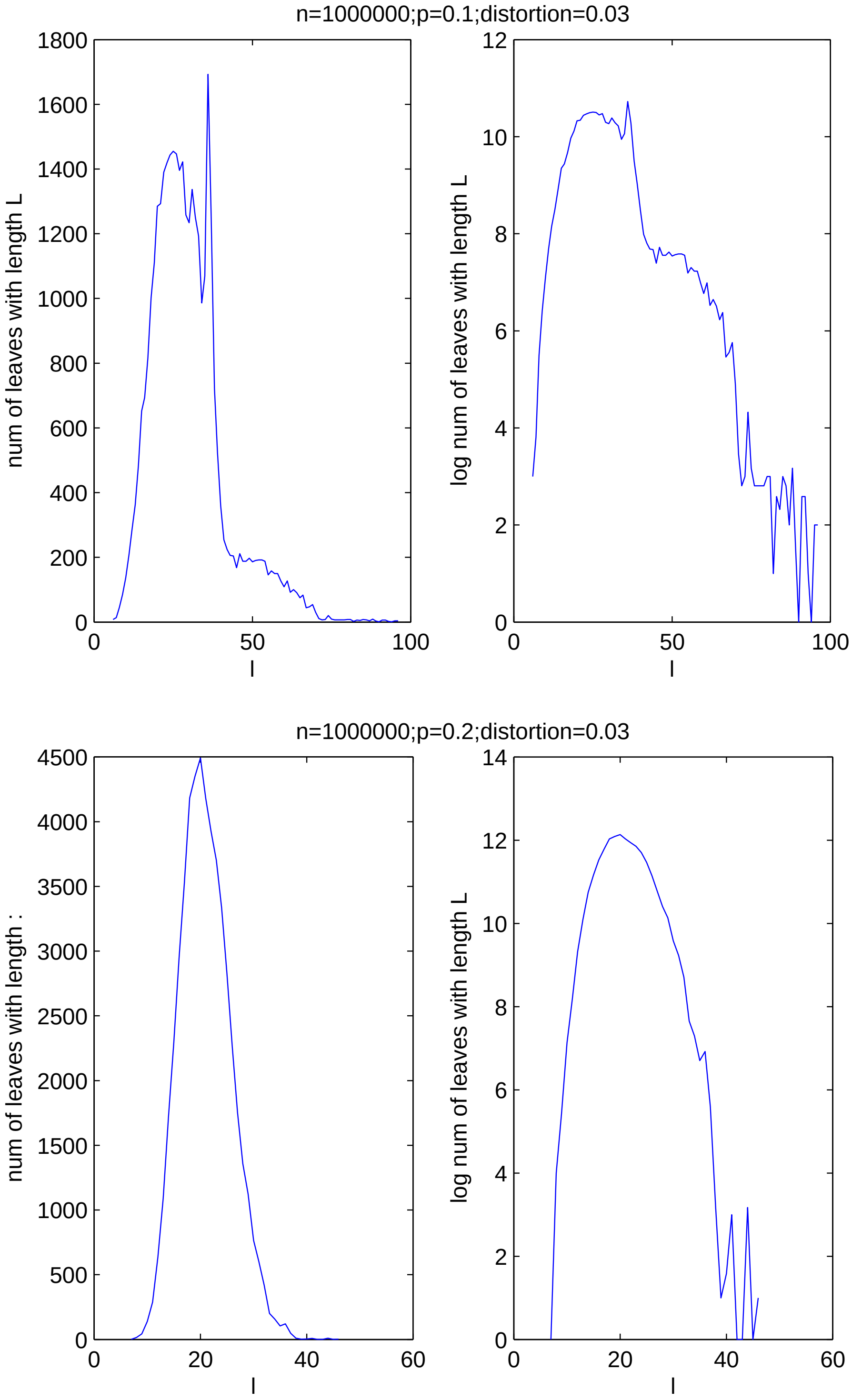


Figure 6.21: Codelet Frequency

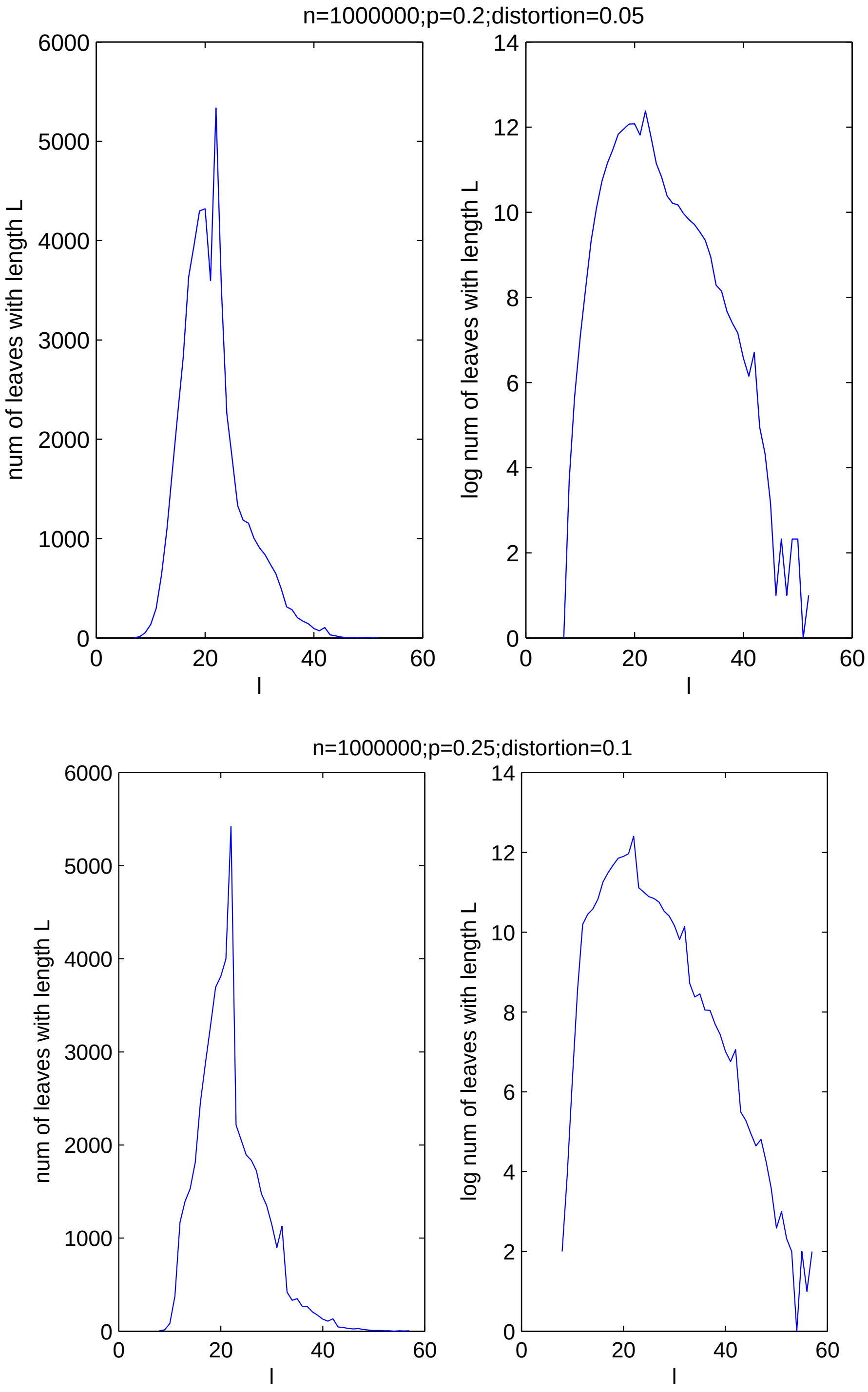


Figure 6.22: The plots show how $|q^* - q_i|$ changes as we parse the sequence.

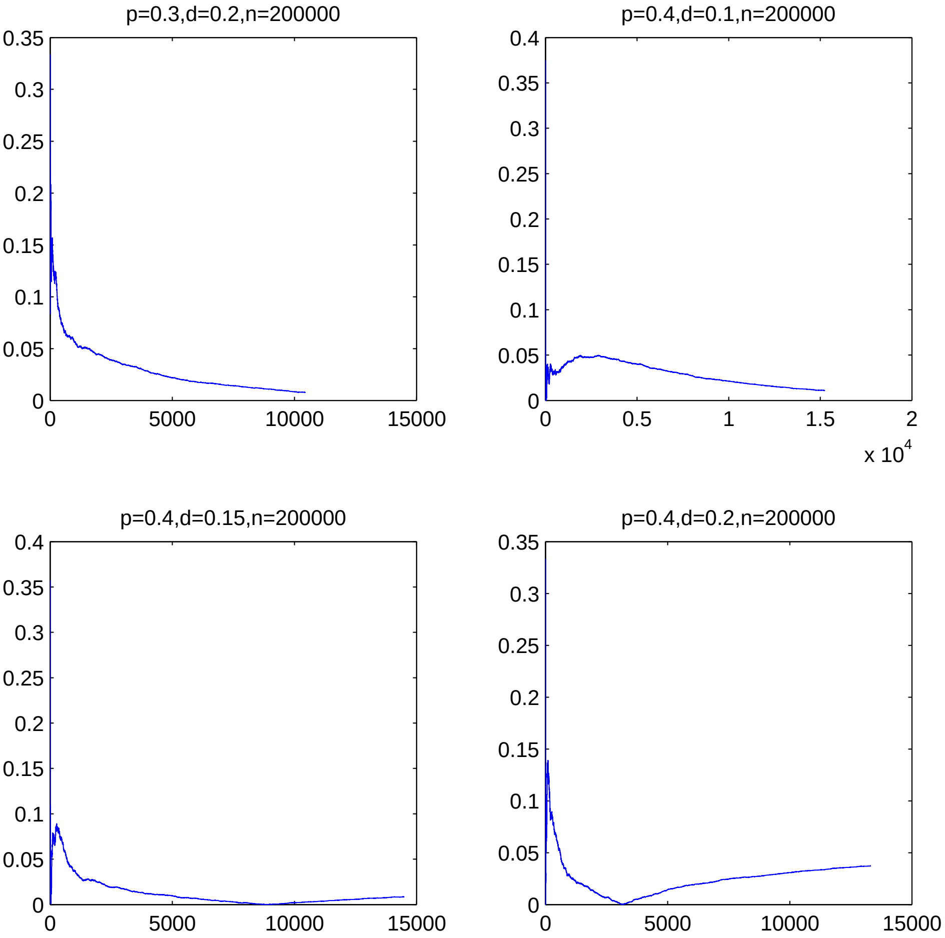


Figure 6.23: The plots show how $|q^* - q_i|$ changes as we parse the sequence.

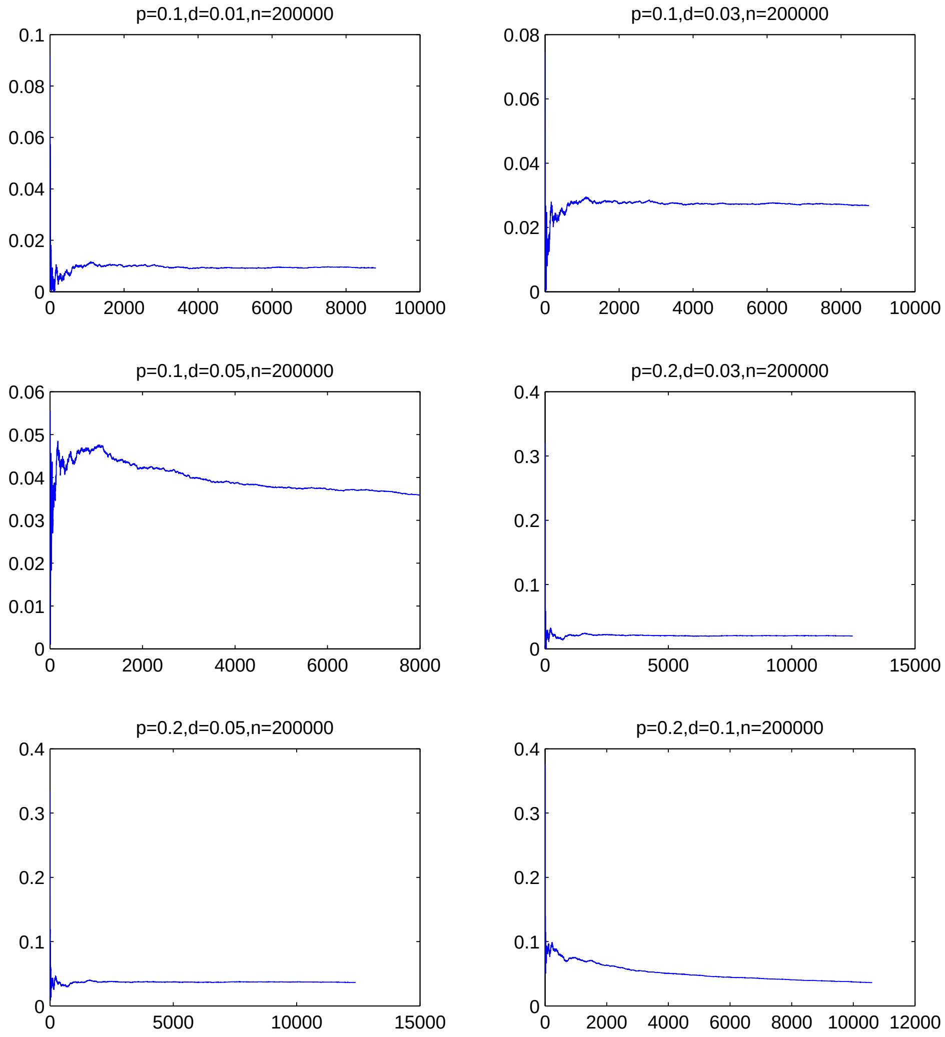


Table 6.12: Achieved distortion

n	p	d	nLeafMost	TimeMost(s)	diffPos
10000	0.5	0.1	1174	0	193
100000	0.3	0.1	7526	22	6095
100000	0.3	0.01	7932	13	0
100000	0.1	0.03	4778	14	336
100000	0.2	0.03	6747	12	0
100000	0.3	0.03	7932	13	0
100000	0.1	0.05	4434	102	2311
100000	0.2	0.05	6731	14	798
100000	0.3	0.05	7938	14	35
100000	0.1	0.05	4434	104	2311
100000	0.5	0.2	7927	85	14851
100000	0.5	0.3	7379	465	23481
100000	0.5	0.4	6337	3017	32679
100000	0.375	0.1	8186	18	6099
100000	0.375	0.2	7043	212	15613
100000	0.375	0.3	5650	1506	24588
100000	0.25	0.05	7424	14	237
100000	0.25	0.1	6826	42	6259
100000	0.25	0.2	4852	1541	16142
1000000	0.1	0.02	36666	535	791
1000000	0.2	0.03	52647	566	173
1000000	0.1	0.01	36708	576	0

Table 6.13: Achieved distortion for "Most Match" method

n	p	d	nLeafMost	TimeMost	diffPos	compRate	dHat	hp-hdHat	optRate
100000	0.1	0.05	4434	96	2311	0.581492	0.02311	0.3104	0.1826
400000	0.3	0.1	25010	255	26289	0.976029	0.06572	0.5315	0.4123
100000	0.1	0.03	4778	13	336	0.631756	0.00336	0.4365	0.2746
400000	0.1	0.03	15978	160	2310	0.597729	0.005775	0.4177	2746
100000	0.3	0.03	7932	13	0	1.10679	0	0.8813	0.6869
400000	0.3	0.03	27320	133	0	1.07488	0	0.8813	0.6869
400000	0.1	0.05	14355	1696	11250	0.531468	0.0281	0.2842	0.1826
100000	0.5	0.2	7927	85	14851	1.10602	0.14851	0.3939	0.2781
100000	0.5	0.3	7379	445	23481	1.02193	0.23481	0.2137	0.1187
100000	0.375	0.2	7043	201	15613	0.970664	0.15613	0.3295	0.2325
200000	0.375	0.2	12755	841	32075	0.933588	0.160375	0.3192	0.2325

6.5.2.5 Heat Plots

Table 6.13 shows a set of simulation results using most match method. Below is the description for each column:

- "diffPos" number of different positions between original sequence and reconstructed one.
- "compRate" compression rate of the algorithm.
- "dHat" is the "diffPos"/n.
- "hp-hdHat" is the $h(p) - h(\text{dHat})$.
- "optRate" is the optimal rate which is $h(p) - h(d)$.

6.5.2.6 Distortion Monitoring

Figures 6.24 and 6.25 show how the distortion of the sequence evolves as we parse it. The x axis shows the length of the parsed sequence, and the y axis shows the total distortion of it. For example, for $n = 100000, p = 0.1$ and $d = 0.05$, when we parsed half of the sequence the total distortion is around 0.02 and when we finish parsing the sequence the total distortion is 0.025 which is closer to desired distortion 0.05.

Figure 6.24: Distortion Monitoring

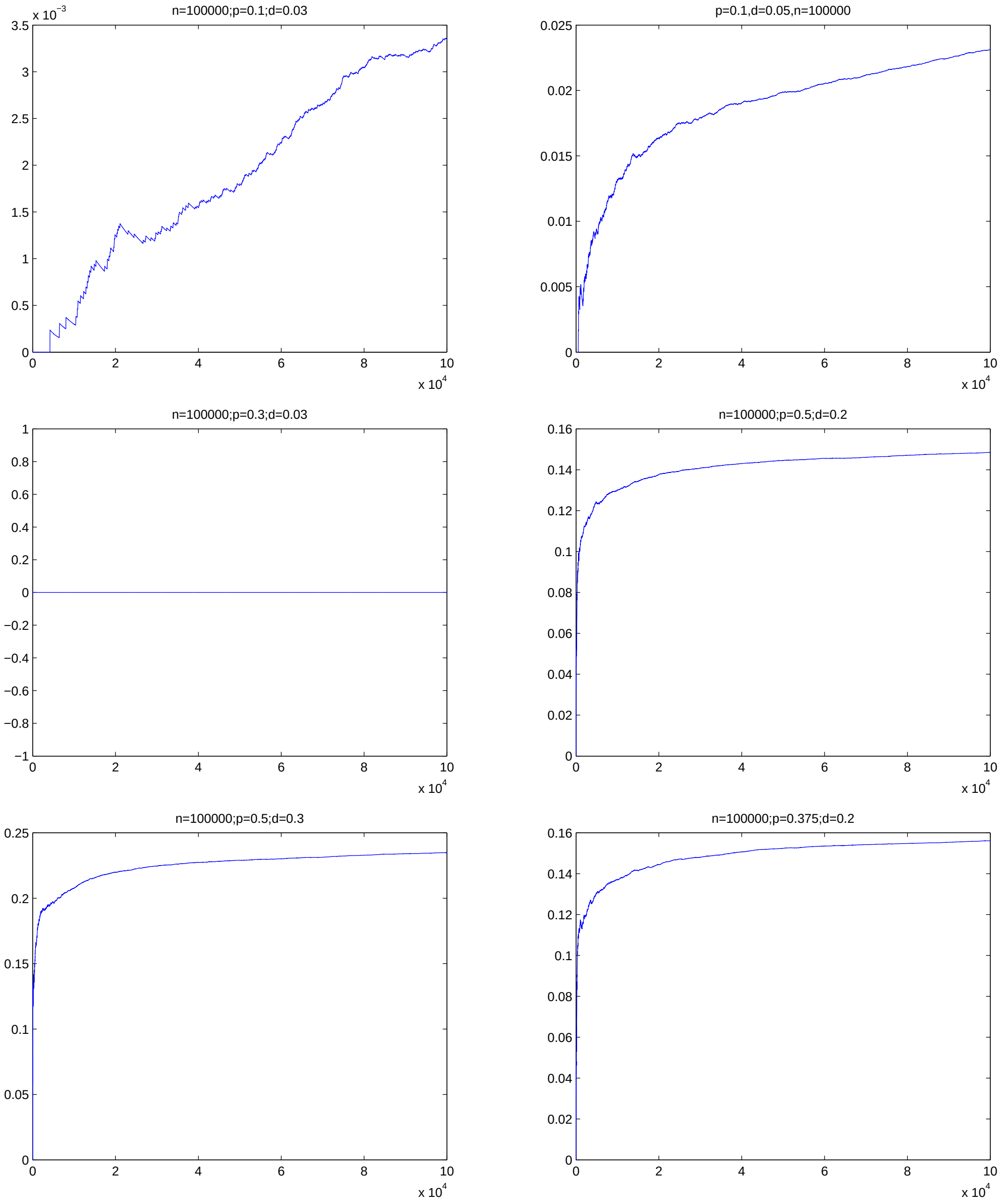
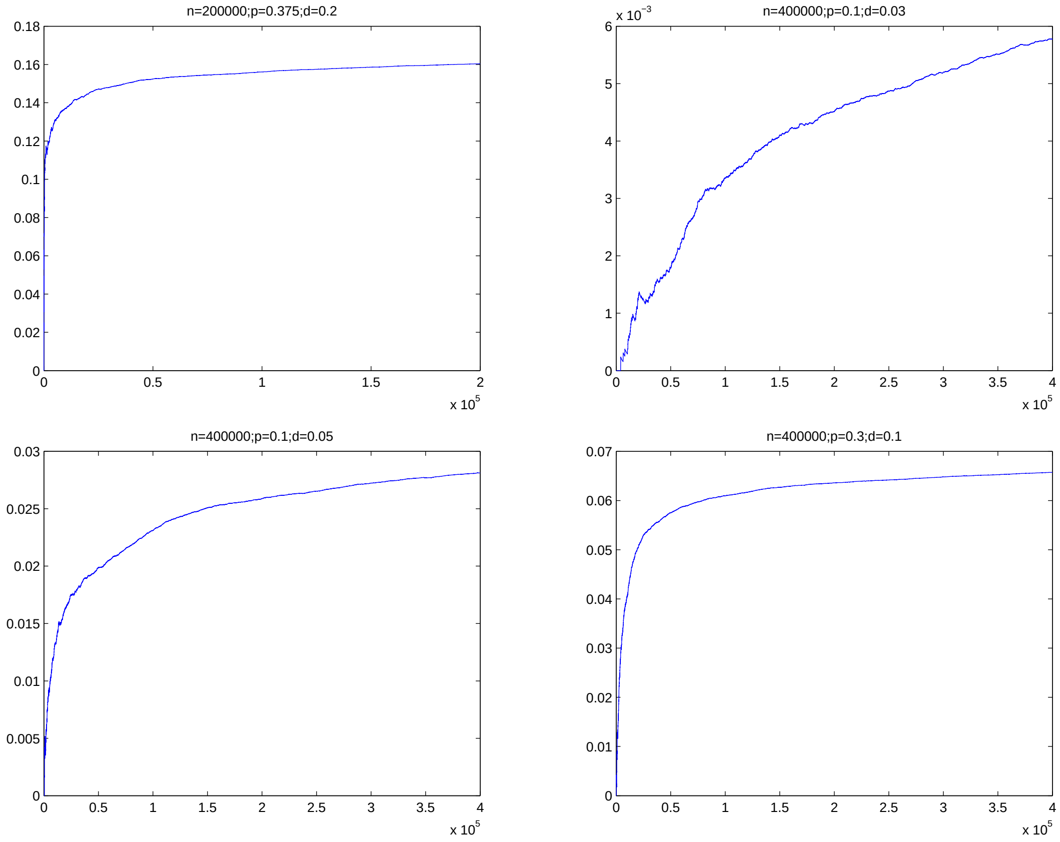


Figure 6.25: Distortion Monitoring



6.5.2.7 Codelet Parsing as an Iterative algorithm

If the compression rate is always decreasing it will finally converge to a value (although not necessarily the optimal value). If a convergence rate be always decreasing then it is easy to see than the length of the parsed sequence in step i is $i \log i^{1+\epsilon}$ for some $\epsilon > 0$. Figures 6.26 to 6.28 show the compression rate vs iteration number and Figures 6.29 and 6.30 shows the length of the parsed phrase.

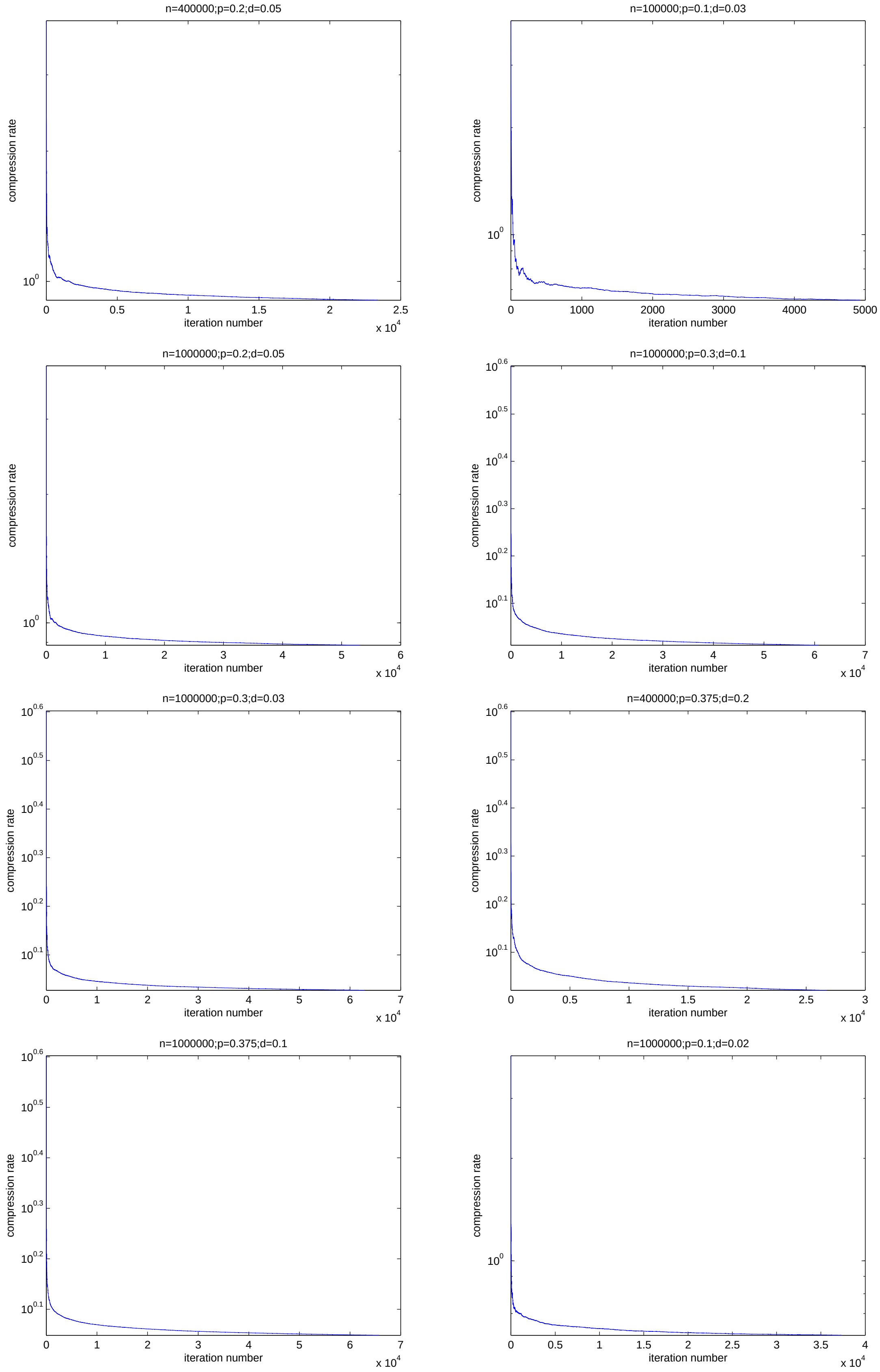


Figure 6.26: Compression Rate vs Iteration number

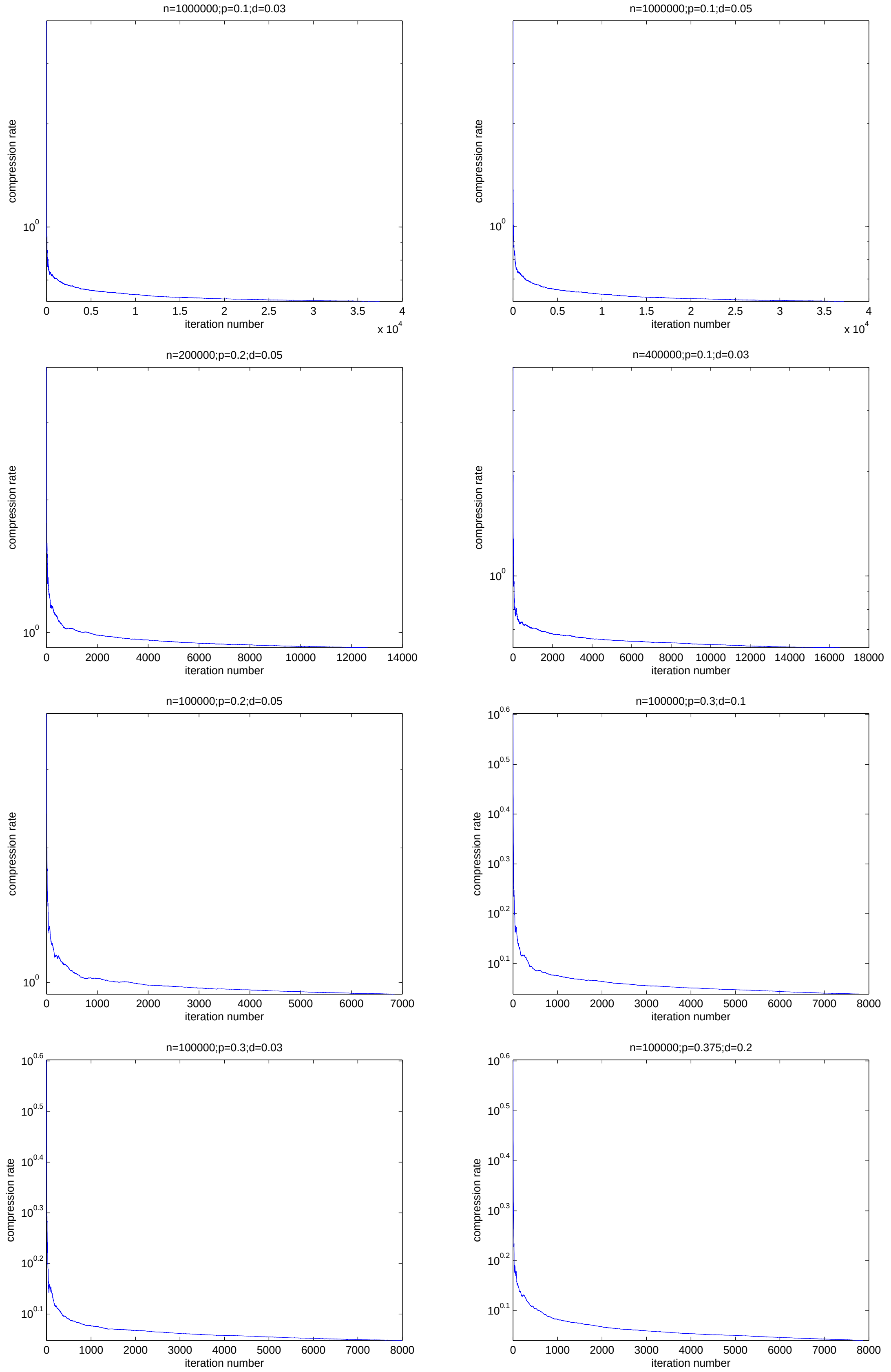


Figure 6.27: Compression Rate vs Iteration number

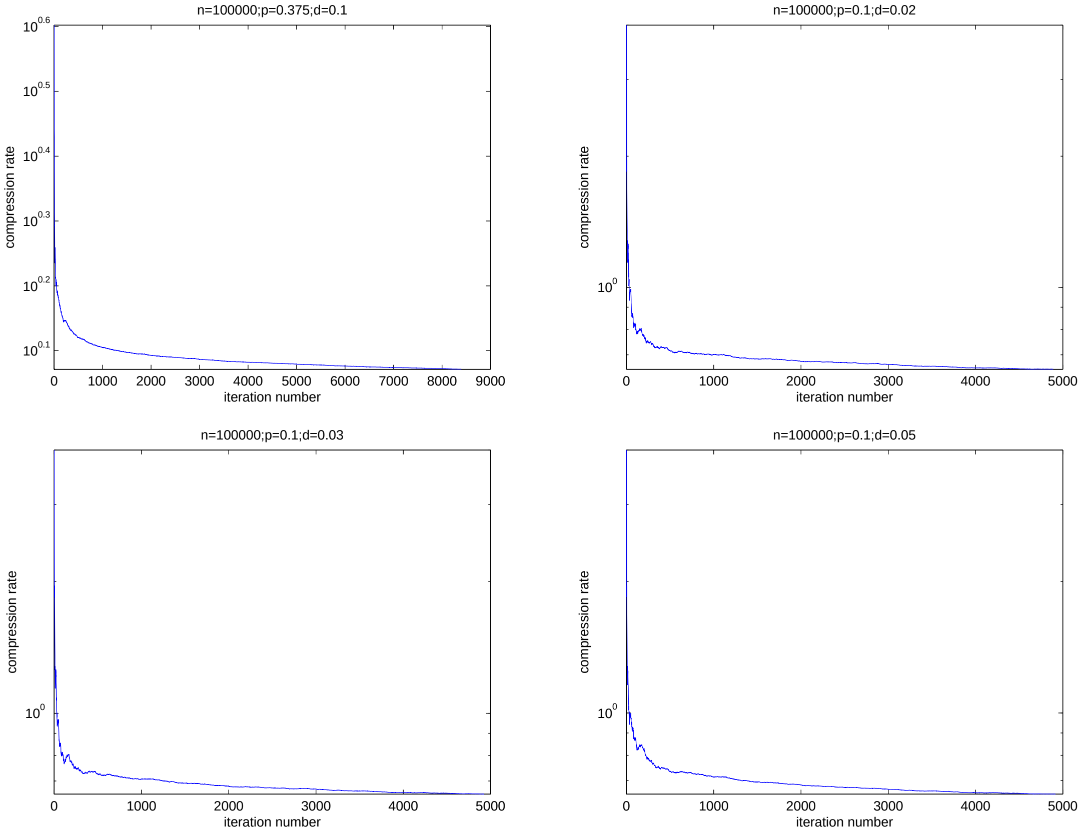


Figure 6.28: Compression Rate vs Iteration number

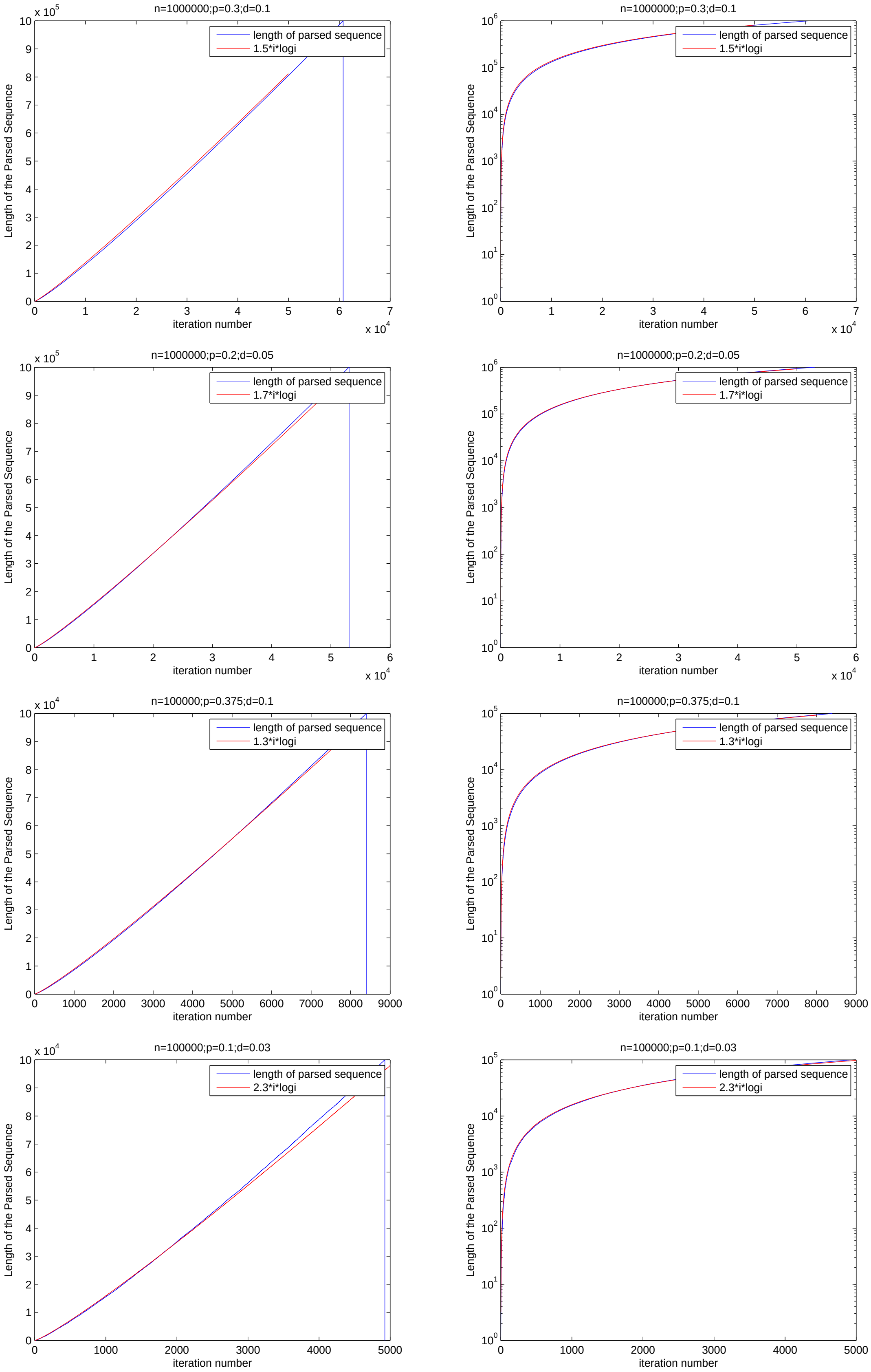


Figure 6.29: length of the parsed sequence vs iteration number

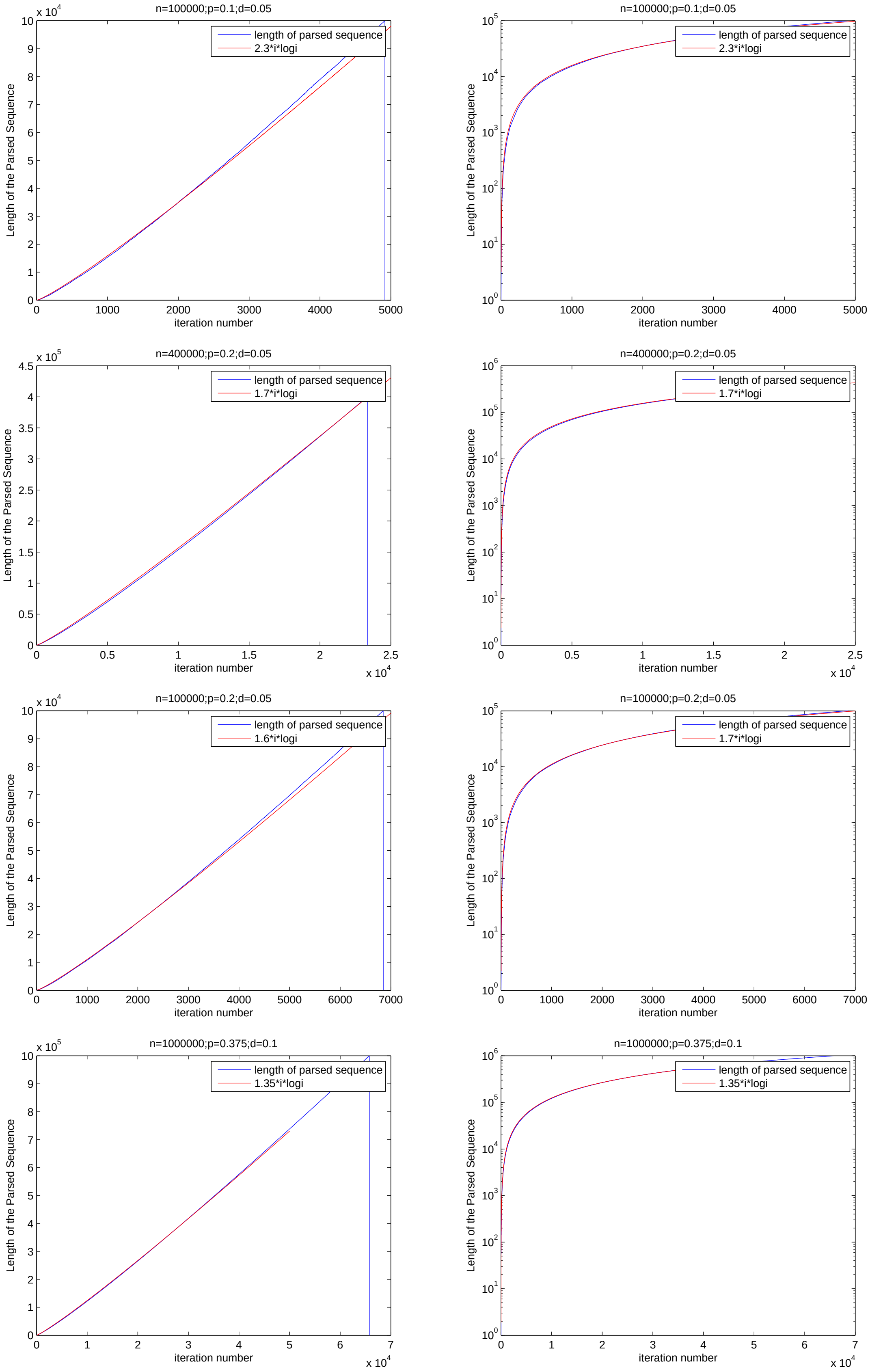


Figure 6.30: length of the parsed sequence vs iteration number

Part III

Rate-Distortion Formulation of a Problem in Cyber-Physical Systems

7

Designing Optimal Watermark Signal for a Stealthy Attacker

7.1 Introduction

Cyber-physical system security is by now a well-motivated and popular problem (see, e.g. [65–68] and the references therein). One typical formulation of the problem considers plants being controlled remotely. An intruder or an attacker is able to change data transmitted across one or both of sensor-controller and controller-actuator channels. The intruder may have constraints in terms of powers, number of components she can act on and so on. The general problem is to design strategies for intruder/attacker to gain information and degrade the performance of the plant maximally and for the sensor/controller to maintain a guaranteed level of performance in spite of the attacker being present.

We consider the attacker acting on the controller-actuator channel. The two works closest to the problem we consider are [68] and [66]. In [68], the authors presented an information theoretic study of the performance degradation that is achievable by an attacker for whom the only constraint is its desire to remain undetected. When the controller is not constrained to conduct any particular detection test, [68] characterized the largest possible estimation error covariance that can be induced by an attacker while remaining stealthy.

However, [68] assumed that the nominal control signal generated by the controller was known to the attacker. As [66] showed, if this assumption does not hold, then the controller can use a watermarking strategy for signaling the presence of an attacker to itself. Specifically, the controller can intentionally add a noisy signal to the nominal control input to detect if an attacker is present. Obviously, adding a noisy signal to the optimal control input degrades the performance of the system. In [66], the authors posed the problem of designing the watermark signal for stationary Gaussian processes to maximize the Kullback-Leibler distance between the compromised and noisy control inputs in the case of a replay attack and proposed a particular (although suboptimal) solution.

We consider the same framework as that in [68] but remove the assumption of the attacker having access to the nominal control input. As mentioned above, this introduces the possibility for the controller to watermark its input. However, differently from [66], we do not limit the attacker to a replay attack. The only constraint we place on the attacker is stealthiness. Intuitively, stealthiness is defined as the difference or distance between the nominal and the corrupted signal that characterizes the difficulty with which a detector can detect whether an attack is in progress. We use mutual information for measuring the distance. Mutual Information (MI) is an information theoretic metric proposed by Shannon which characterizes the amount of information that one random variable can provide about another random variable [44]. After presenting a notion of stealthiness in terms of MI, the main contribution of this work is twofold. First, we consider the problem of identifying the watermark signal that minimizes the similarity (as measured by MI) between the watermarked control input and the control input as corrupted by the attacker, while the degradation in the LQG performance as compared to the nominal case is bounded. We show that the optimal watermark when an attacker is replacing the control input with a values that are realizations of a Gaussian random variable is a Gaussian signal. Further, this watermark can be obtained by solving a Semi-Definite Programming (SDP) problem. Then, we consider the problem of designing the control input that the attacker should substitute that is optimal in the sense that it is as similar as possible to the watermarked control input (for stealthiness), while being as dissimilar as possible to the nominal control input (so that the performance degradation is maximized). We show that if the controller is using a Gaussian watermark signal, the attacker should introduce a control input that is distributed according to a Gaussian random variable. We consider only the one-step version of the problem.

This chapter is organized as follows. Section 7.2 presents the system model and the problem formulation. In Section 7.3, we prove that the best watermarking signal for a Gaussian attacker is given by a Gaussian random variable. In Section 7.4, we prove that the worst attack for a Gaussian watermark is also Gaussian.

Notation: A sequence of variables $\{x_0, x_2, \ldots, x_N\}$ is denoted by x_0^N or simply as x^N if the lower and upper limits are clear from the context. If $N \to \infty$ we denote the infinite sequence by x_0^∞. $\mathcal{N}(\mu, \sigma^2)$ refers to a random variable with a Gaussian pdf with mean μ and variance σ^2. $M > N$ (respectively, $M \geq N$) for matrices M and N implies that $M - N$ is positive definite (respectively, positive semidefinite).

Let f and g be two probability measures on the same measurable space. Let df/dg be the Radon-Nikodym derivative of f with respect to g [69]. The Kullback-Leiber distance $D(f||g)$

between f and g is defined as:

$$D(f||g) = \int (\log \frac{df}{dg}) df, \quad \text{if } \frac{df}{dg} \text{ exists.}$$

The mutual information between two random variable X and Y is defined as:

$$I(X;Y) = D(f_{X,Y}||f_X \times f_Y),$$

where $f_X \times f_Y$ denotes the product measure. $E[X]$ denotes the expectation of random variable X.

7.2 Problem Formulation

System Model We consider a time invariant process described by:

$$x_{t+1} = ax_t + u_t + w_t,$$

$$y_t = cx_t + \nu_t.$$

where $x_t \in \mathbb{R}$ is the state at time t and $y_t \in \mathbb{R}$ is the sensor observation. The process noise sequence $w_1^\infty \sim \mathcal{N}(0, \sigma_\omega^2)$ and the measurement noise sequence $\nu_1^\infty \sim \mathcal{N}(0, \sigma_\nu^2)$ are white noise sequences. The initial condition $x_0 \sim \mathcal{N}(0, P_0)$ is independent of these noise sequences. The control input $u_t \in \mathbb{R}$ is the output of a pre-designed LQG controller.

To detect if an attacker is present, the controller may want to change the control input u_t to a watermarked version u_t^*. In the sequel, we consider u_t^* to be the control input transmitted by the controller, with $u_t = u_t^*$ if no watermarking is performed.

Attack Model The attacker has access to the communication channel from the controller to the actuator. The attacker can replace any control input u_t^* by an input of its choice $\tilde{u}_t$.

To design this attack signal, we assume that the attacker has access to the system parameters $(a, c, \sigma_\nu^2, \sigma_\omega^2, P_0)$, measurements y_0^t and the nominal control inputs u_0^t. However, even though the attacker knows that watermarking may be performed, it does not know the watermarked control inputs u_t^*.

Remark We focus on the one step case. In other words, we consider the case when $t = 0$. For simplicity, we thus remove the subscript t for all the signals. Extension of the work to multi-stage case is an interesting problem but beyond the scope of this work. □

Problem Statement To motivate the problem statement, let us consider the problem from the point of view of the controller. The controller wants to design the best possible watermark, i.e., it wants to design a watermark signal u^* so that the similarity between the watermark signal u^* and the attacker signal $\tilde{u}$ is minimized. We propose the use of mutual information as the similarity metric between u^* and $\tilde{u}$. At the same time, adding noise to the nominal control input degrades the performance and the controller wants to minimize this degradation. As a proxy for the LQG performance, we consider the constraint $E\|u - u^*\|^2 < \epsilon$. Thus the problem from the point of view of the controller is given by

$$\begin{aligned} &\underset{p(u^*|u)}{\text{minimize}} && I(\tilde{u}; u^*) \\ &\text{subject to} && E\|u - u^*\|^2 < \epsilon. \end{aligned} \tag{7.1}$$

Now, let us consider the problem from the point of view of the attacker. The attacker aims to generate an attack signal $\tilde{u}$ that is as similar as possible to the watermarked control input u^* to remain undetected or stealthy. Once again, to capture the notion of stealthiness, we propose the use of mutual information as the similarity metric between u^* and $\tilde{u}$. At the same time, the attacker wants to substitute the nominal control input u by a signal $\tilde{u}$ that is as dissimilar as possible to the nominal control input so that the performance of the process is degraded maximally. Thus the problem that attacker is interested in is given by:

$$\underset{p(\tilde{u}|u),\, I(\tilde{u};u)=I_u}{\text{maximize}} \quad I(\tilde{u};u^*). \tag{7.2}$$

The main result of this chapter is that (i) if $\tilde{u}$ is distributed according to a Gaussian random variable (i.e., for a Gaussian attacker), the optimal solution of (7.1) is a Gaussian watermarking signal; and (ii) if u^* is distributed according to a Gaussian random variable (i.e., for a Gaussian watermarking signal), the optimal solution of (7.2) is a Gaussian attacker.

Remark Note that the problem (7.1) is formally similar to the sequential rate-distortion problem in [70]. However, the problem in [70] considers the same two signals in the optimization objective and the constraint. In our problem, the optimization objective is the MI between $\tilde{u}$ and u^*, while the constraint is in terms of the variables u and u^*. Similarly, while the problem formulation in (7.1) is similar to privacy-accuracy tradeoff problem in [71], but the solution here is completely different than the one presented in [71]. The problem (7.2) is also formally similar to the one considered in [72] and our proof follows similar principles. □

7.3 Optimal Watermark for a Gaussian Attacker

In this section, we assume that the attacker's policy is Gaussian, i.e., $p(\tilde{u}|u)$ can be written in the form of[1]

$$\tilde{u} = \Sigma_{21}\Sigma_{11}^{-1}u + \xi, \;\; \xi \sim \mathcal{N}(0, \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}) \tag{7.3}$$

[1] The results in this subsection will be stated under general setting in which random variables u, $\tilde{u}, u^*$ are multi-dimensional.

where ξ is independent of u, and that the joint distribution $p(u, \tilde{u})$ is a zero-mean Gaussian distribution with a covariance matrix

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \triangleq \mathbb{E}_p \begin{bmatrix} u \\ \tilde{u} \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \end{bmatrix}^\top . \tag{7.4}$$

Under this assumption, we next observe that an optimal watermarking policy for (7.1) is also Gaussian.

Theorem 37. There exists an optimal policy $p(u^*|u)$ for (7.1) that can be written in the form of

$$u^* = \Sigma_{31}\Sigma_{11}^{-1}u + \eta, \quad \eta \sim \mathcal{N}(0, \Sigma_{33} - \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{13}) \tag{7.5}$$

with some matrix Σ_{31}, Σ_{33} such that $\Sigma_{33} - \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{13} \succeq 0$, where η is independent of u.

Proof Suppose $p(u^*|u)$ is an arbitrary feasible watermarking policy for (7.1) such that the value of the objective function is c. It is sufficient to prove that there always exists another feasible watermarking policy $p'(u^*|u)$ in the form of (7.5) such that the value of the objective function is $c' \leq c$.

To this end, let $p(u, u^*) \triangleq p(u^*|u)p(u)$ be a joint distribution induced by a given watermarking policy $p(u^*|u)$, and without loss of generality, assume $p(u, u^*)$ is zero-mean. Let

$$\begin{bmatrix} \Sigma_{11} & \Sigma_{13} \\ \Sigma_{31} & \Sigma_{33} \end{bmatrix} \triangleq \mathbb{E}_p \begin{bmatrix} u \\ u^* \end{bmatrix} \begin{bmatrix} u \\ u^* \end{bmatrix}^\top . \tag{7.6}$$

be the covariance matrix. Notice that if $p(u^*|u)$ is given, a joint distribution

$$p(u, \tilde{u}, u^*) \triangleq p(u^*|u)p(\tilde{u}|u)p(u)$$

is also determined, and its covariance matrix is

$$\Sigma \triangleq \mathbb{E}_p \begin{bmatrix} u \\ \tilde{u} \\ u^* \end{bmatrix} \begin{bmatrix} u \\ \tilde{u} \\ u^* \end{bmatrix}^\top = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{13} \\ \Sigma_{31} & \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{12} & \Sigma_{33} \end{bmatrix}. \tag{7.7}$$

We have used the fact that

$$\mathbb{E}_p \tilde{u} u^{*\top} = \mathbb{E}_p(\Sigma_{21}\Sigma_{11}^{-1}u + \xi)u^{*\top} = \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{13}.$$

Now, consider an alternative policy $p'(u^*|u)$ defined by

$$u^* = \Sigma_{31}\Sigma_{11}^{-1}u + \eta, \quad \eta \sim \mathcal{N}(0, \Sigma_{33} - \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{13})$$

where η is independent of u, and denote the induced joint distribution by

$$p'(u, \tilde{u}, u^*) \triangleq p'(u^*|u)p(\tilde{u}|u)p(u).$$

Clearly, $p(u, \tilde{u}, u^*)$ and $p'(u, \tilde{u}, u^*)$ share the same covariance matrix (7.7). Thus, if $p(u^*|u)$ is a feasible policy (i.e., $\mathbb{E}_p\|u - u^*\|^2 < \epsilon$) then $p'(u^*|u)$ is also feasible, since $\mathbb{E}_p\|u - u^*\|^2 = \mathbb{E}_{p'}\|u - u^*\|^2$ follows from the fact that $p(u, \tilde{u}, u^*)$ and $p'(u, \tilde{u}, u^*)$ have the same covariance matrix. Moreover,

$$
\begin{aligned}
&I_p(\tilde{u}; u^*) - I_{p'}(\tilde{u}; u^*) && (7.8)\\
&= \int \log \frac{dp_{\tilde{u},u^*}}{d(p_{\tilde{u}} \times p_{u^*})} dp_{\tilde{u},u^*} - \int \log \frac{dp'_{\tilde{u},u^*}}{d(p'_{\tilde{u}} \times p'_{u^*})} dp'_{\tilde{u},u^*}\\
&= \int \log \frac{dp_{u^*|\tilde{u}}}{dp_{u^*}} dp_{\tilde{u},u^*} - \int \log \frac{dp'_{u^*|\tilde{u}}}{dp'_{u^*}} dp'_{\tilde{u},u^*}\\
&= \int \log \frac{dp_{u^*|\tilde{u}}}{dp_{u^*}} dp_{\tilde{u},u^*} - \int \log \frac{dp'_{u^*|\tilde{u}}}{dp'_{u^*}} dp_{\tilde{u},u^*} && (7.9)\\
&= \int dp_{\tilde{u},u^*} \log \frac{dp_{u^*|\tilde{u}}}{dp'_{u^*|\tilde{u}}}\\
&= \int \left(\int \log \frac{dp_{u^*|\tilde{u}}}{dp'_{u^*|\tilde{u}}} dp_{u^*|\tilde{u}} \right) dp_{\tilde{u}}\\
&= \int D(dp_{u^*|\tilde{u}} || dp'_{u^*|\tilde{u}}) dp_{\tilde{u}} \geq 0
\end{aligned}
$$

where (7.9) follows from the fact that $\int \log \frac{dp'_{u^*|\tilde{u}}}{dp'_{u^*}} dp'_{\tilde{u},u^*} = \int \log \frac{dp'_{u^*|\tilde{u}}}{dp'_{u^*}} dp_{\tilde{u},u^*}$, since $\log \frac{dp'_{u^*|\tilde{u}}}{dp'_{u^*}}$ is a quadratic function and p and p' have the same second order moment. Thus we have constructed a Gaussian policy $p'(u^*|u)$ attaining the objective value $c' \leq c$. □

Our second result presents a computationally efficient method to synthesize an optimal watermark policy. In particular, we show that optimal watermark policy can be obtained by solving a semidefinite program. Note that from Theorem 37, to characterize the optimal watermark policy, we need to specify Σ_{13} and Σ_{33}.

Theorem 38. Suppose (7.3) and (7.4) are fixed. Then, an optimal watermark policy is given by (7.5), where Σ_{13} and Σ_{33} are obtained as the optimal solution to a determinant-maximization problem

$$
\begin{aligned}
&\underset{\Pi \succ 0, \Sigma_{31}, \Sigma_{33}}{\text{minimize}} \quad \frac{1}{2} \log \det \Sigma_{22} - \frac{1}{2} \log \det \Pi\\
&\text{subject to} \begin{cases} \Sigma \succeq 0, \quad \text{Trace}(\Sigma_{11} - \Sigma_{31} - \Sigma_{13} + \Sigma_{33}) \leq \epsilon \\ \begin{bmatrix} \Sigma_{22} - \Pi & \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{13} \\ \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{12} & \Sigma_{33} \end{bmatrix} \succeq 0. \end{cases}
\end{aligned}
\tag{7.10}
$$

Proof We can write the objective function in (7.1) as

$$\begin{aligned}
I(\tilde{u}; u^*) &= h(\tilde{u}) - h(\tilde{u}|u^*)\\
&= \frac{1}{2}\log\det\Sigma_{22} - \frac{1}{2}\log\det(\Sigma_{22} - \Sigma_{23}\Sigma_{33}^{-1}\Sigma_{32})\\
&= \begin{cases} \underset{\Pi}{\text{minimize}} & \frac{1}{2}\log\det\Sigma_{22} - \frac{1}{2}\log\det\Pi\\ \text{subject to} & 0 < \Pi \preceq \Sigma_{22} - \Sigma_{23}\Sigma_{33}^{-1}\Sigma_{32}\end{cases}\\
&= \begin{cases} \underset{\Pi\succ 0}{\text{minimize}} & \frac{1}{2}\log\det\Sigma_{22} - \frac{1}{2}\log\det\Pi\\ \text{subject to} & \Sigma_{22} - \Pi - \Sigma_{23}\Sigma_{33}^{-1}\Sigma_{32} \succeq 0\end{cases}\\
&= \begin{cases} \underset{\Pi\succ 0}{\text{minimize}} & \frac{1}{2}\log\det\Sigma_{22} - \frac{1}{2}\log\det\Pi\\ \text{subject to} & \begin{bmatrix}\Sigma_{22} - \Pi & \Sigma_{23}\\ \Sigma_{32} & \Sigma_{33}\end{bmatrix} \succeq 0,\end{cases}\\
&= \begin{cases} \underset{\Pi\succ 0}{\text{minimize}} & \frac{1}{2}\log\det\Sigma_{22} - \frac{1}{2}\log\det\Pi\\ \text{subject to} & \begin{bmatrix}\Sigma_{22} - \Pi & \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{13}\\ \Sigma_{31}\Sigma_{11}^{-1}\Sigma_{12} & \Sigma_{33}\end{bmatrix} \succeq 0.\end{cases}
\end{aligned}$$

Where $h(.)$ is the differential entropy [44] and Σ_{23} is the cross-correlation between $\tilde{u}$ and u^*. The constraint in (7.1) can be written in terms of the components of the matrix Σ as $\text{Trace}(\Sigma_{11} - \Sigma_{31} - \Sigma_{13} + \Sigma_{33}) \leq \epsilon$. Combined, we obtain (7.10). □

7.4 Worst Attacker for a Gaussian Watermark Signal

We now present the result complementary to Theorem 37 by showing that the solution of the optimization problem in (7.2) is also a Gaussian signal. In this optimization problem, the attacker aims to maximize the performance degradation while retaining its stealthiness. Thus, it wants to design an attack signal that is as similar to the watermark signal to make sure that the attack $\tilde{u}$ is undetectable, and at the same time, it wishes to ensure that the similarity between the nominal control input and attacker doesn't exceed an specific level

I_u.

Theorem 39. The optimal solution $\tilde{u}$ of the optimization problem in (7.2) is a Gaussian random variable.

Proof The proof technique follows from [72] and [73]. We obtain a lower bound for $I(\tilde{u}; u|u^*)$ and show that the lower bound is achieved if and only if u and $\tilde{u}$ are jointly Gaussian. To this end, note that

$$\begin{aligned} I(\tilde{u}; u|u^*) &= I(\tilde{u}; u, u^*) - I(\tilde{u}; u^*) \\ &= I(\tilde{u}; u) + I(\tilde{u}; u^*|u) - I(\tilde{u}; u^*) \\ &= I(\tilde{u}; u) - I(\tilde{u}; u^*), \end{aligned} \tag{7.11}$$

where (7.11) is obtained by using the fact that $\tilde{u} \to u \to u^*$ is a Markov chain. Given that $I(\tilde{u}; u) = I_u$, maximizing $I(\tilde{u}; u^*)$ is equivalent to minimizing $I(\tilde{u}; u|u^*)$.

Also, since u and u^* are jointly Gaussian, we can write $u^* = au + \xi$, where ξ is a Gaussian random variable that is independent of u. Since ξ is also independent of $\tilde{u}$, we can use the entropy power inequality to write

$$\begin{aligned} e^{2h(u^*|\tilde{u})} &\geq e^{2h(au|\tilde{u})} + e^{2h(\xi|\tilde{u})} \\ &= e^{2h(u|\tilde{u})}|a|^2 + e^{2h(\xi|\tilde{u})} \\ &= e^{2h(u|\tilde{u})}|a|^2 + e^{2h(\xi)} \\ &= e^{2h(u)-2I(\tilde{u};u)}|a|^2 + e^{2h(\xi)} \\ &= e^{2h(u)}e^{-2I(\tilde{u};u)}|a|^2 + e^{2h(\xi)} \\ &= (2\pi e\Sigma_u)e^{-I(\tilde{u};u)}|a|^2 + 2\pi e\Sigma_\xi, \end{aligned} \tag{7.12}$$

where Σ_ξ denotes the covariance of ξ. Note that the above inequality holds with equality if and only if u and $\tilde{u}$ are jointly Gaussian. Now, note that

$$\begin{aligned} e^{2h(u^*|\tilde{u})} &= e^{2h(u^*)-2I(\tilde{u};u^*)} \\ &= 2\pi e\Sigma_{u^*}e^{-I(\tilde{u};u^*)}. \end{aligned} \tag{7.13}$$

Thus, substituting (7.12) in (7.13) we have

$$\begin{aligned} 2\pi e\Sigma_{u^*}e^{-I(\tilde{u};u^*)} &\geq 2\pi e\Sigma_u e^{-I(\tilde{u};u)} \\ &\qquad |a|^2 + 2\pi e\Sigma_\xi \\ \Rightarrow -I(\tilde{u};u^*) \ &\geq \log\Bigg((2\pi e\Sigma_u)e^{-I(\tilde{u};u)} \\ &\qquad |a|^2 + (2\pi e\Sigma_\xi)\Bigg) \\ &\qquad - \log(2\pi e\Sigma_{u^*}), \end{aligned} \tag{7.14}$$

where, once again, the inequality holds with equality if and only if u and $\tilde{u}$ are jointly Gaussian. Finally, substituting (7.14) in (7.11), yields

$$\begin{aligned} I(\tilde{u};u|u^*) \geq I(\tilde{u};u) & \\ &+ \log\Bigg((2\pi e\Sigma_u)e^{-I(\tilde{u};u)}|a|^2 \\ &+ (2\pi e\Sigma_\xi)\Bigg) - \log(2\pi e\Sigma_{u^*}), \end{aligned}$$

with equality if and only if u and $\tilde{u}$ are jointly Gaussian. Now, notice that $I(\tilde{u};u) = I_u$ which is a constant. Thus, $I(\tilde{u};u|u^*)$ is minimized, and consequently $I(\tilde{u};u^*)$ is maximized, if u and $\tilde{u}$ are jointly Gaussian. □

Bibliography

[1] J. Kieffer, “A unified approach to weak universal source coding,” *IEEE Transactions on Information Theory*, vol. 24, no. 6, pp. 674—682, November 1978.

[2] L. Davisson, “Universal noiseless coding,” *IEEE Transactions on Information Theory*, vol. 19, no. 6, pp. 783—795, November 1973.

[3] L. D. R. J. M. M. B. Pursley, M. Wallace, “Efficient universal noiseless source codes,” *IEEE Transactions on Information Theory*, vol. 279, no. 3, pp. 269—279, May 1981.

[4] Q. Xie and A. Barron, “Asymptotic minimax regret for data compression, gambling and prediction,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 431—445, March 2000.

[5] M. Drmota and W. Szpankowski, “Precise minimax redundancy and regrets,” *IEEE Trans. Information Theory*, vol. 50, pp. 2686–2707, 2004.

[6] Y. Shtarkov, T. Tjalkens, and F. Willems, “Multialphabet universal coding of memoryless sources,” *Problems of Information Transmission*, vol. 31, no. 2, pp. 114—127, 1995.

[7] F. M. J. Willems, “The context-tree weighting method: Extensions,” *IEEE Transactions on Information Theory*, vol. 44, pp. 792–798, 1998.

[8] I. Csiszar and P. Shields, “Redundancy rates for renewal and other processes,” *IEEE Transactions on Information theory*, vol. 42, pp. 2065–2072, 1996.

[9] P. Flajolet and W. Szpankowski, “Analytic variations on redundancy rates of renewal processes,” *IEEE Transactions on Information theory*, vol. 48, pp. 2911–2921, 2002.

[10] J. R. M. F. M. J. Weinberger, “A universal finite memory source,” *IEEE Transactions on Information theory*, vol. 41, pp. 643–652, 1995.

[11] N. Santhanam, “Making the correct mistakes,” in *Presentation at new results section at ISIT 2011*, St. Petersburg, 2011.

[12] J. Rosenthal, *A first look at rigorous probability theory*, 2nd ed. World Scientific, 2008.

[13] N. Merhav and M. Feder, “Universal prediction,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2124—2147, October 1998.

[14] A. Orlitsky, N. Santhanam, and J. Zhang, “Universal compression of memoryless sources over unknown alphabets,” *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1469—1481, July 2004.

[15] J. Kingman, “The mathematics of genetic diversity,” *SIAM*, 1980.

[16] S. Zabell, “Predicting the unpredictable,” *Synthese*, vol. 90, pp. 205–232, 1992.

[17] ——, “The continuum of inductive methods revisited,” in *The Cosmos of Science: Essays of Exploration*, J. Earman and J. D. Norton, Eds. Pittsburgh, PA, USA: The University of Pittsburgh Press, 1997, ch. 12.

[18] ——, *Symmetry and Its Discontents: Essays on the History of Inductive Probability*, ser. Cambridge Studies in Probability, Induction, and Decision Theory. Cambridge: Cambridge University Press, 2005.

[19] M. Hosseini and N. Santhanam, “Characterizing the asymptotic per-symbol redundancy of memoryless sources over countable alphabets in terms of single-letter marginals,” *Entropy, 16(7), 4168-4184*, 2014, full version available from arXiv doc id:1404:0062.

[20] S. Boucheron, A. Garivier, and E. Gassiat, "Coding on countably infinite alphabets," Available from arXiv doc id: 0801.2456, 2008.

[21] D. Haussler and M. Opper, "Mutual information, metric entropy and cumulative relative entropy risk," *The Annals of Statistics*, vol. 25, no. 6, pp. 2451–2492, 1997.

[22] S. Boucheron, A. Garivier, and E. Gassiat, "Coding on countably infinite alphabets," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 358–373, 2009.

[23] D. Haussler, "A general minimax result for relative entropy," *IEEE Transactions on Information Theory*, vol. 43, no. 4, pp. 1276–1280, 1997.

[24] D. Bontemps, S. Boucheron, and E. Gassiat, "About adaptive coding on countable alphabets," *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 808–821, 2014.

[25] A. Orlitsky and N. Santhanam, "Lecture notes on universal compression," Available online from `http://www-ee.eng.hawaii.edu/~prasadsn/`.

[26] N. Santhanam, V. Anantharam, A. Kavcic, and W. Szpankowski, "Data driven weak universal redundancy," in *Proceedings of IEEE Symposium on Information Theory*, 2014.

[27] N. Santhanam, "Probability estimation and compression involving large alphabets," Ph.D. dissertation, University of California, San Diego, 2006.

[28] M. Asadi, R. P. Torghabeh, and N. P. Santhanam, "Stationary and transition probabilities in slow mixing, long memory markov processes," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5682–5701, 2014.

[29] I. Csiszár and Z. Talata, "Context tree estimation for not necessarily finite memory processes, via bic and mdl," *IEEE Transactions on Information theory*, vol. 52, no. 3, pp. 1007–1016, 2006.

[30] R. Krichevsky and V. Trofimov, "The preformance of universal coding," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199—207, March 1981.

[31] Q. Xie and A. Barron, "Minimax redundancy for the class of memoryless sources," *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 647—657, March 1997.

[32] W. Szpankowski, "On asymptotics of certain recurrences arising in universal coding," *Problems of Information Transmission*, vol. 34, no. 2, pp. 142—146, 1998.

[33] V. K. Trofimov, "Redundancy of universal coding of arbitrary markov sources," *Problemy Peredachi Informatsii*, vol. 10, no. 4, pp. 16–24, 1974.

[34] L. Davisson, "Minimax noiseless universal coding for markov sources," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 211–215, 1983.

[35] K. Atteson, "The asymptotic redundancy of bayes rules for markov chains," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2104–2109, 1999.

[36] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 40—47, January 1996.

[37] P. Jacquet and W. Szpankowski, "Markov types and minimax redundancy for markov sources," *IEEE Transactions on Information Theory*, vol. 50, no. 7, pp. 1393–1402, 2004.

[38] F. M. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[39] J. Rissanen, "A universal data compression system," *IEEE Transactions on information theory*, vol. 29, no. 5, pp. 656–664, 1983.

[40] F. M. Willems, "The context-tree weighting method: Extensions," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 792–798, 1998.

[41] M. Falahatgar, A. Orlitsky, V. Pichapati, and A. T. Suresh, "Learning markov distri-

butions: Does estimation trump compression?" in *Information Theory (ISIT), 2016 IEEE International Symposium on.* IEEE, 2016, pp. 2689–2693.

[42] A. K. Dhulipala and A. Orlitsky, "Universal compression of markov and related sources over arbitrary alphabets," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4182–4190, 2006.

[43] I. Csiszár, P. C. Shields *et al.*, "Information theory and statistics: A tutorial," *Foundations and Trends® in Communications and Information Theory*, vol. 1, no. 4, pp. 417–528, 2004.

[44] T. Cover and J. Thomas, *Elements of Information Theory.* John Wiley and sons., 1991.

[45] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problemy Peredachi Informatsii*, vol. 23, no. 3, pp. 3–17, 1987.

[46] K. Oshiro, C. Wu, and N. Santhanam, "Jackknife estimation for markov processes with no mixing constraints," in *Information Theory (ISIT), 2017 IEEE International Symposium on.* IEEE, 2017, pp. 3020–3024.

[47] C. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv*, vol. 7, no. 4, pp. 142–163, 1959.

[48] H. Morita and K. Kobayashi, "An extension of lzw coding algorithm to source coding subject to a fidelity criterion," in *4th Joint Swedish- Soviet Int. Workshop on Information Theory*, 1989.

[49] Y. ESteinberg and M. Gutman, "An algorithm for source coding subject to a fidelity criterion, based on string matching. information theory," *IEEE Transactions on Information theory*, vol. 39, no. 3, pp. 877–886, 1993.

[50] G. Louchard and W. Szpankowski, "On the average redundancy of lempel ziv code," *IEEE Transactions on Information Theory*, vol. 43, no. 1, pp. 1–7, 1997.

[51] E. H. Yang and J. C. Kieffer, "On the performance of data compression algorithms based upon string matching," *IEEE Transactions on Information theory*, vol. 44, no. 1, pp. 47–65, 1998.

[52] R. Zamir and K. Rose, "Natural type selection in adaptive lossy compression," *IEEE Transactions on Information theory*, vol. 47, no. 1, pp. 99–111, 2001.

[53] C. F. Atallah, M. J. and P. Dumas, "A randomized algorithm for approximate string matching," *Algorithmica*, vol. 29, no. 3, pp. 468–486, 2001.

[54] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[55] E. H. Yang and J. C. Kieffer, "Simple universal lossy data compression schemes derived from the lempel-ziv algorithm," *IEEE Transactions on Information theory*, vol. 42, no. 1, pp. 239–245, 1996.

[56] J. C. Kieffer, "A survey of the theory of source coding with a fidelity criterion," *IEEE Transactions on Information theory*, vol. 39, no. 5, pp. 1473–1490, 1993.

[57] a. W. T. Jalali, S., "Rate-distortion via markov chain monte carlo," in *In Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008.

[58] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Transactions on Information theory*, vol. 56, no. 4, pp. 1751–1768, 2010.

[59] J. Muramatsu, "Variable-length lossy source code using a constrained-random-number generator." information theory," *IEEE Transactions on Information theory*, vol. 61, no. 6, pp. 3574–3592, 2015.

[60] D. Modha, "Codelet parsing: Quadratic-time, sequential, adaptive algorithms for lossy compression," in *Proceedings of the Data Compression Conference*, 2003.

[61] N. Santhanam and D. Modha, "Lossy compression algorithms for memoryless sources," Information Theory and Applications, Feb 2011.

[62] D. Modha and N. Santhanam, “Making the correct mistakes,” in *Proceedings of the Data Compression Conference*, 2006.

[63] ——, “Making the correct mistakes,” in *Proceedings of the Data Compression Conference*, 2006.

[64] S. Zaks and N. Dershowitz, “The cycle lemma and some applications,” *Europ. J. of Combinatorics*, 1990.

[65] H. Sandberg, S. Amin, and K. Johansson, “Cyberphysical security in networked control systems: An introduction to the issue,” *IEEE Control Systems Magazine*, vol. 35, 2015.

[66] Y. Mo, S. Weerakkody, and B. Sinopoli, “Physical authentication of control systems designing watermarked control inputs to detect counterfeit sensor outputs,” *IEEE Control Systems Magazine*, pp. 93–109, 2015.

[67] Y. Mo, R. Chabukswar, and B. Sinopoli, “Detecting integrity attacks on scada systems,” *IEEE Transactions on Control Systems Technology*, pp. 1396–1407, 2014.

[68] C. Z. Bai, F. Pasqualetti, and V. Gupta, “Security in stochastic control systems: Fundamental limitations and performance bounds,” in *American Control Conference*, 2015, pp. 195–200.

[69] G. B. Folland, *Real analysis: modern techniques and their applications.* John Wiley and sons, New York, 2013.

[70] T. Tanaka, K. K. K. Kim, P. A. Parrilo, and S. K. Mitter, “Semidefinite programming approach to Gaussian sequential rate-distortion trade-offs,” *arXiv preprint arXiv:1411.7632*, 2014.

[71] F. P. Calmon and N. Fawaz, “Privacy against statistical inference,” in *In 50th Annual Allerton Conference on Communication, Control, and Computing*, 2012, pp. 1401–1408.

[72] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss., “Information bottleneck for Gaussian variables,” *Journal of Machine Learning Research*, Nov 2005.

[73] T. Berger and R. Zamir, "A semi-continuous version of the Berger-Yeung problem," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1520–1526, 1999.