Technische Universität München
Lehrstuhl für Nachrichtentechnik
Prof. Dr. sc. techn. Gerhard Kramer

Master's Thesis

Algorithms for Simulation of Discrete Memoryless Sources

Vorgelegt von:
Rana Ali Amjad

München, October 29, 2013

Betreut von:
Dr.-Ing. Georg Böcherer

Master's Thesis am
Lehrstuhl für Nachrichtentechnik (LNT)
der Technischen Universität München (TUM)
Titel : Algorithms for Simulation
of Discrete Memoryless Sources
Autor : Rana Ali Amjad

Rana Ali Amjad
Schröfelhofstrasse 14-07-06
81375 München
ranaali.amjad@tum.de

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, 29.10.2013

Ort, Datum (Rana Ali Amjad)

Abstract

Simulation of Discrete Memoryless Sources, alternatively known as approximate random number generation, has attracted a lot of interest in recent years due to its new found application in various information theoretic problems including information theoretic security, coordination and probabilistic shaping for reliable data transmission. Unlike channel coding or source coding, developing practical codes for approximate random number generation is still largely an open problem. The two important parameters for the design of such codes are rate and distortion. The problem of random number generation has been studied in the literature from various perspectives including an information theoretic perspective and an algorithmic perspective, but there exists no unified approach for the development and analysis of practical codes. In this thesis our aim is to develop the required framework and propose practical codes for approximate random number generation. We have studied the following two variants of the random number generation problem

- Distribution matching.
- Resolution coding for target distributions.

We start this thesis by extending the framework of rooted trees with probabilities for the analysis of variable length codes designed for approximate random number generation. This is the framework we use for the rest of the thesis for developing various practical codes for the two variants of the random number generation problem. We propose a bound on the difference between the entropy rates of variable length codes based on rooted trees and the entropy of a memoryless source. The bound is in terms of normalized informational divergence. We call this result the Entropy-Divergence Theorem. It is employed several times in later chapters to prove converses and to show achievability for various codes developed.

The first variant of random number generation we study is distribution matching. It deals with the reversible generation of random numbers according to some target distribution from a fair bit stream. A converse defining an upper bound on the maximum achievable rates is proved. We continue by proposing zero error block-to-block (b2b) and fixed-to-variable length (f2v) codes for distribution matching and show their asymptotic

optimality, i.e, they achieve the rate upper bound. Moreover we propose an ϵ-error b2b matcher which combines the better rate distortion performance of zero error variable length codes with the simplicity of a b2b code at the expense of a small error probability ϵ. The code construction is such that it allows for a nice tradeoff between rate, divergence, error probability and complexity of the code. This encoder also achieves the rate upper bound.

Resolution coding for target distributions is the second variant of simulation of DMS studied in this thesis. In this case the focus is on the deterministic transformation of a fair bit stream to generate random numbers according to some target distribution. A general converse is proved, extending the lower bound on rate earlier proposed in the literature for b2b encoders to make it applicable to any variable length encoder. An optimal b2b encoder is proposed and it is shown to asymptotically achieve the lower bound. Later we develop f2v encoders and variable-to-fixed length (v2f) encoders that significantly improve the rate-distortion performance in the finite block length regime.

Although the focus of the discussion in this thesis is on approximate random number generation, in the process of developing the codes we also solved some informational divergence optimization problems over discrete sets. Moreover we also characterize the set of distributions that corresponds to a special case of Finite State Generators.

Contents

List of Figures

List of Tables

1 Introduction

Simulation of a Discrete Memoryless Source (DMS), alternatively known as approximate random number generation problem, has been studied in literature from both information theoretic and algorithmic complexity perspectives. It refers to the problem of transforming a DMS P_B into a target DMS P_Y where $P_B = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$. The two important parameters in the study of approximate random number generation is the distortion which indicates how good is the approximation and rate which indicates the number of input symbols required per output symbol by the encoder/algorithm.

There are many variants of this problem found in literature inspired from various applications. A classical direction of work is exact random number generation where the approximating process is exactly the same as target process. This has originally been discussed in [1] from an algorithmic complexity perspective where the authors have suggested the rate optimal algorithm for exact random number generation. Suboptimal algorithms for transforming any arbitrary DMS into another arbitrary DMS have been presented in [2]. These works are mainly inspired from the application of approximate random number generation to generate input data for system model simulation.

In [3], Han and Verdu have analyzed the problem of random process simulation at the output of a channel. This has recently found its application in information theoretic secrecy and coordination. Approximate random number generation is a special case of this setup when the channel is identity and target random process is a product distribution. We refer to this special case as Resolution coding for target distribution. Steinberg and Verdu[4] have developed a rate-distortion theory for random process simulation for identity channel. In both of these works, the distortion is either characterized by total variational distance or normalized informational divergence between the approximating process and the target process. In [5], Hou and Kramer have strengthened the results presented in [3] for special case of Discrete Memoryless Channel (DMC) and product target distributions by using un-normalized informational divergence as distortion measure.

A more recent variant of approximate random number generation has been discussed in [6] where we require the process of random number generation to be reversible. It is called Distribution Matching. This finds its applications in probabilistic shaping for Discrete Noiseless Channels and Discrete Memoryless Channels.

What is lacking in the aforementioned literature is a discussion on the design of practical codes for these problems.For exact random number generation, the algorithms presented in [1] and [2] lead to codebooks with infinite length codewords which is not suitable for practical implementation unless the codebook has special structure. For approximate random number generation, although the results are proved in [3, 4] and [5] using finite length block codes but the analysis is mostly information theoretic in nature using random coding arguments for showing existence of encoders. In [6], Böcherer has not considered rate for the discussion of encoder design for distribution matching. Moreover only v2f encoders are considered for the task.
In this thesis, we provide a unified approach for studying different variants of the random number generation problem from both information theoretic and algorithmic perspective. We will propose information theoretic bounds for the suggested problems and then present practical algorithms that can achieve these bounds asymptotically. We also evaluate the performance of these algorithms in the finite length regime. In the process of building this unified framework, we also fill the gaps mentioned earlier regarding the existing literature. Our main contributions in this thesis are

- Using the rooted trees with probabilities framework we have derived a bound on the difference between the entropy rates of variable length codes based on rooted trees and a memoryless source in terms of normalized informational divergence. The bound is then used in later sections to derive converses for distribution matching, resolution coding for target distributions and exact random number generation. Moreover it has also been used for various proposed codes to show achievability results.

- For distribution matching, we prove a converse that establishes an upper bound on the achievable rates. Construction of optimal one-to-one b2b matcher is discussed. Then we derive an efficient algorithm to calculate the distortion optimal zero error f2v matcher. The f2v distribution matcher asymptotically achieves the upper bound on achievable rates. Moreover we construct practical ϵ-error b2b matcher. For this matcher we can easily control the tradeoff between rate, probability of error, distortion and complexity. This matcher also asymptotically achieves the upper bound on achievable rates.

- For resolution coding, we first generalize the converse presented in [7] to variable length encoders. This converse establishes a lower bound on the achievable rates. Then we propose a unified framework to cover both the information theoretic and algorithmic complexity perspective to analyze the problem. Using this framework we construct various b2b codes and variable length codes and show that they achieve

the rate lower bound asymptotically. We also assess the performance of these algorithms in the finite length regime.

2 Preliminaries

2.1 Properties of Probability Distributions

Let X be a random variable with probability distribution P_X on some finite set $\mathcal{X}$. Similarly define Y with P_Y on $\mathcal{Y}$. Denote by P_{XY} some joint distribution on $\mathcal{X} \times \mathcal{Y}$ having marginal distributions P_X and P_Y on $\mathcal{X}$ and $\mathcal{Y}$ respectively.

Definition 1. *For any positive integer M, P_X is M-type iff*

$$P_X(x) \in \left\{ \frac{1}{M}, \frac{2}{M}, \cdots, \frac{M}{M} \right\} \qquad \forall x \in \text{supp}(P_X) \tag{2.1}$$

where $\text{supp}(P_X)$ denotes the subset of $\mathcal{X}$ s.t. $P_X(x) > 0 \quad \forall x \in \text{supp}(P_X)$. Note that if P_X is M-type, then it is also KM-type for any positive integer K. Moreover if P_X is irrational, i.e., $P_X(x)$ is irrational for some $x \in \mathcal{X}$, then P_X is not M-type for any finite integer value of M.

Definition 2. *The resolution $\mathsf{R}(P_X)$ of a probability distribution P_X is the minimum $\log M$ such that P_X is M-type.*

log refers to logarithm with base 2 in this thesis unless explicitly stated otherwise. If P_X is not M-type for any value of M then $R(P_X) = \infty$. In particular, the resolution of irrational distributions is equal to infinity.

Definition 3. *P_X is dyadic if*

$$P_X(x) = 2^{-k_x} \qquad k_x \in \mathbb{Z}_{\geq 0} \quad \forall x \in \text{supp}(P_X) \tag{2.2}$$

where $Z_{\geq 0}$ represents the set of non negative integers.

2.2 Information Theoretic Quantities and Identities

Definition 4. *The entropy of a random variable X is*

$$\mathbb{H}(X) = \mathbb{H}(P_X) = \sum_{x \in \text{supp}(P_X)} P_X(x)[-\log P_X(x)]. \tag{2.3}$$

It is also sometimes called uncertainty in a random variable X. Important properties of entropy are

- **Non-negativity:** $\mathbb{H}(P_X) \geq 0$.
- **Maximum Entropy Lemma:** For a given finite set $\mathcal{X}$, $\mathbb{H}(X) \leq \log|\operatorname{supp}(P_X)| \leq \log|\mathcal{X}|$ and the equalities are obtained iff X is uniformly distributed on $\operatorname{supp}(P_X)$ and $\mathcal{X}$ respectiveley.
- **Conditional entropy:** The conditional entropy of X given Y is

$$\mathbb{H}(P_{X|Y}) = \mathbb{H}(X|Y) = \sum_{y\in\mathcal{Y}} P_Y(y)\,\mathbb{H}(P_{X|Y=y}) \tag{2.4}$$

$$= \sum_{y\in\mathcal{Y}} P_Y(y) \sum_{i\in\operatorname{supp}(P_{X|Y=y})} P_{X|Y=y}(x)[-\log P_{X|Y=y}(x)]. \tag{2.5}$$

  Conditioning does not increase entropy:

$$\mathbb{H}(X|Y) \leq \mathbb{H}(X). \tag{2.6}$$

- **Chain rule:** The entropy of P_{XY} can be evaluated as

$$\mathbb{H}(P_{XY}) = \mathbb{H}(XY) = \mathbb{H}(X) + \mathbb{H}(Y|X) \tag{2.7}$$

$$= \mathbb{H}(Y) + \mathbb{H}(X|Y). \tag{2.8}$$

Definition 5. *Informational divergence, also known as Kullback-Leibler divergence or relative entropy, of two probability distributions P_X and P'_X is defined as*

$$\mathbb{D}(P_X\|P'_X) = \sum_{x\in\operatorname{supp}(P_X)} P_X(x)\log\frac{P_X(x)}{P'_X(x)}. \tag{2.9}$$

Some important properties of informational divergence are stated below

- **Non-negativity** $\mathbb{D}(P_X\|P'_X) \geq 0$, with equality iff $P_X = P'_X$.
- **Non symmetric** $\mathbb{D}(P_X\|P'_X)$ is not symmetric, i.e., $\mathbb{D}(P_X\|P'_X) \neq \mathbb{D}(P'_X\|P_X)$. Hence it is not a metric.

Definition 6. *Mutual information between two random variables X and Y is defined as*

$$\mathbb{I}(X;Y) = \mathbb{D}(P_{XY}\|P_XP_Y). \tag{2.10}$$

Some properties of mutual information are:

- **Non-negativity** $\mathbb{I}(X;Y) \geq 0$.
- **Expansion in entropy terms** Mutual information can be expanded as

$$\mathbb{I}(X;Y) = \mathbb{H}(X) - \mathbb{H}(X|Y) \tag{2.11}$$
$$= \mathbb{H}(Y) - \mathbb{H}(Y|X). \tag{2.12}$$

  From these alternative expressions we can see that mutual information measures the average reduction of the uncertainty in one random variable knowing another random variable. Hence it is a measure of dependence of two random variables on eachother. Based on these alternative expression mutual information, is also denoted by $\mathbb{I}(P_X, P_{Y|X})$ and $\mathbb{I}(P_Y, P_{X|Y})$.

- **Symmetry:** Mutual information is symmetric in its arguments, in contrast to informational divergence, i.e., $\mathbb{I}(X;Y) = \mathbb{I}(Y;X)$.

2.3 Fano's Inequality

Theorem 1 (Fano's Inequality)**.** *We estimate X by observing Y. Denote by $\hat{X} = f(Y)$ our estimate. Define $P_e = \Pr(X \neq \hat{X})$. Then*

$$\mathbb{H}(P_e) + P_e \log |\mathcal{X}| \geq \mathbb{H}(X|\hat{X}) \geq \mathbb{H}(X|Y) \tag{2.13}$$

Hence if given Y we still have some uncertainty left in X then the probability of making an error in estimation is bounded away from zero.

2.4 Information Channels

2.4.1 Discrete Memoryless Channel

A *Discrete Memoryless Channel* (DMC) from $\mathcal{X}$ to $\mathcal{Y}$ is defined by a conditional probability distribution $P_{Y|X}$. A Conditional distribution can be represented by an $m \times n$ matrix $\boldsymbol{H}$ where $|\mathcal{X}| = n$ and $|\mathcal{Y}| = m$. For a probability distribution P_X on $\mathcal{X}$, which we call the channel input distribution, the distribution over the channel output symbols $\mathcal{Y}$ is

$$P_Y = \boldsymbol{H} P_X. \tag{2.14}$$

The capacity of a DMC is calculated by maximizing the mutual information between X and Y where the optimization is over the channel input distribution P_X. Hence

$$\mathsf{C} = \max_{P_X} \mathbb{I}(P_X, P_{Y|X}). \tag{2.15}$$

2.5 Rooted Trees With Probabilities

We consider finite complete trees with finite alphabets $\mathcal{Y} = \{0, 1, \ldots, m-1\}$.

2.5.1 Trees

2.5.1.1 Tree-Based Definition

A *complete* tree $\mathcal{T}$ consists of branching nodes $\mathcal{B}$ with m successors and leaves $\mathcal{L}$ with no successors. Each node except the root node has exactly one predecessor. Each branch is labelled by a symbol from the alphabet $\mathcal{Y}$ and for each branching node, each label from the alphabet is the label of *exactly* one outgoing edge. Each node is uniquely identified by the string of labels on the path from the root to the node. The root node is identified by the empty string ε.

2.5.1.2 Set-Based Definition

A prefix free complete set $\mathcal{L}$ over an alphabet $\mathcal{Y}$ is a set of strings with letters in $\mathcal{Y}$ such that each right-infinite string over $\mathcal{Y}$ starts with exactly one string in $\mathcal{L}$. Let $\mathcal{B}$ be the set of all prefixes of strings in $\mathcal{L}$ including the empty string ϵ. The union $\mathcal{T} = \mathcal{L} \cup \mathcal{B}$ is called a complete tree.
For a complete tree $\mathcal{T}$, we will denote the set of branching nodes and the set of leaves by $\mathcal{B}(\mathcal{T})$ and $\mathcal{L}(\mathcal{T})$ respectively. For notational convenience, we will write $\mathcal{B} = \mathcal{B}(\mathcal{T})$ and $\mathcal{L} = \mathcal{L}(\mathcal{T})$ if the referred tree is clear from the context. Furthermore dictionary and codebook in the context of this thesis are rooted trees.

2.5.2 Probabilities

2.5.2.1 Probabilities induced by leaf distribution

We can define a probability distribution P_T on $\mathcal{L}(\mathcal{T})$. P_T can be extended to $\mathcal{T}$ by associating with each string $t \in \mathcal{B}(\mathcal{T})$, a probability

$$P_T(t) = \sum_{u \in \mathcal{L}\colon\, u = t*} P_T(u). \tag{2.16}$$

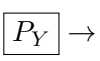

Figure 2.1: Discrete Memoryless Source P_Y.

i.e., the sum of the probabilities of all elements of $\mathcal{L}$ having the same prefix t. Furthermore, for each string $t \in \mathcal{B}$, we define a branching distributions P_{Y_t} on $\mathcal{Y}$ by

$$P_{Y_t}(y) = \frac{P_T(ty)}{P_T(t)}. \qquad \forall y \in \mathcal{Y} \tag{2.17}$$

This branching distribution is basically a conditional distribution over $\mathcal{Y}$ for the next letter in the sequence given the string t is the occured prefix.
Note that whenever we refer to P_T as a probability distribution, it will refer to the probability distribution formed by P_T on $\mathcal{L}(\mathcal{T})$.

2.5.2.2 Probabilities induced by alphabet distribution

Let P_Y be a distribution on $\mathcal{Y}$. Then we associate with each string $t = y_1 \cdots y_{\ell(t)} \in \mathcal{T}$ a probability

$$P_Y^{\mathcal{T}}(t) = P_Y(y_1) \cdots P_Y(y_{\ell(t)}) \tag{2.18}$$

where $\ell(t)$ denotes the number of branches on the path from the root to the node t. This induces a distribution on $\mathcal{L}(\mathcal{T})$ and we denote it by $P_Y^{\mathcal{L}(\mathcal{T})}$. For each $t \in \mathcal{L}(\mathcal{T})$, we have

$$P_Y^{\mathcal{L}(\mathcal{T})}(t) = P_Y^{\mathcal{T}}(t) = P_Y(y_1) \cdots P_Y(y_{\ell(t)}). \tag{2.19}$$

For notational convenience, we write $P_Y^{\mathcal{L}} = P_Y^{\mathcal{L}(\mathcal{T})}$ when the tree $\mathcal{T}$ is clear from the context.

2.6 Discrete Memoryless Source

A DMS generating symbols from some finite alphabet $\mathcal{Y}$ according to some source distribution P_Y is depicted in Fig. 2.1. Since the distribution P_Y completely characterizes the DMS, we refer to it by DMS P_Y. The probability of a string $x = y_1 y_2 \cdots y_{\ell(x)}$, , $y_i \in \mathcal{Y}$, appearing at the ouput of DMS P_Y is $\prod_{i=1}^{\ell(x)} P_Y(y_i)$.
Denote by $\mathcal{X}$ a complete $|\mathcal{Y}|$-ary tree. Any right infinite sequence generated by the DMS P_Y must have one of the strings $x \in \mathcal{L}(\mathcal{X})$ as the prefix. Using P_Y as the branching distribution at every branching node in $\mathcal{B}(\mathcal{X})$ induces the leaf distribution $P_Y^{\mathcal{L}(\mathcal{X})}$ on

$\mathcal{L}(\mathcal{X})$ such that the probability of a string $x \in \mathcal{L}(\mathcal{X})$ being generated by the DMS P_Y is equal to $P_Y^{\mathcal{L}(\mathcal{X})}(x)$.A DMS P_Y generates symbols from alphabet $\mathcal{X}$ according to the probability distribution $P_Y^{\mathcal{L}(\mathcal{X})}$.

2.7 Important Algorithms and Codes

2.7.1 Tunstall Coding

Tunstall Coding is a method to construct a rooted tree with probabilities for a given alphabet distribution P_Y such that the rooted tree has certain properties.
Suppose we want to construct a complete $|\mathcal{Y}|$-ary tree with N leaves. For a complete $|\mathcal{Y}|$-ary tree $N-1$ must be an integer multiple of $|\mathcal{Y}|-1$. We start with an extended root node such that it has $|\mathcal{Y}|$ leaves. Denote this tree by $\mathcal{T}_1$ and initialize $i=1$. For each value of i denote by $\mathcal{T}_i$ the complete $|\mathcal{Y}|$-ary tree we have in the ith step. We define an alphabet induced leaf distribution $P_Y^{\mathcal{L}_i} = P_Y^{\mathcal{L}(\mathcal{T}_i)}$ on $\mathcal{L}(\mathcal{T}_i)$ using P_Y as alphabet distribution. Choose the leaf $t \in \mathcal{L}_i$ such that $P_Y^{\mathcal{L}_i}(t) \geq P_Y^{\mathcal{L}_i}(s) \quad \forall s \in \mathcal{L}_i$. Make this leaf t into a branching node by extending it to have $|\mathcal{Y}|$ children. Increment i and repeat the same procedure again until we have $\mathcal{L}(\mathcal{T}_i) = N$.
Denote by $\mathcal{X}$ the output tree of Tunstall Coding. Some of the properties of $\mathcal{X}$ are as follows

- Since we extend the leaf which has highest induced leaf probability according to P_Y at each step, $\mathcal{X}$ maximizes the following expression among all the possible $|\mathcal{Y}|$-ary trees $\mathcal{T}$ with N leaves.
$$\sum_{x \in \mathcal{B}(\mathcal{X})} P_Y^{\mathcal{X}}(t) \geq \sum_{t \in \mathcal{B}(\mathcal{T})} P_Y^{\mathcal{T}}(t) \tag{2.20}$$
- By the Path Length Lemma [8, Sec. 2.2.2], we have
$$\mathbb{E}[\ell(T)] = \sum_{t \in \mathcal{B}(\mathcal{T})} P_Y^{\mathcal{T}}(t) \tag{2.21}$$
and based on the previous property we know that $\mathcal{X}$ maximizes the R.H.S of (2.21). Hence it maximizes $\mathbb{E}[\ell(T)]$ over the set of all complete $|\mathcal{Y}|$-ary complete trees with N leaves. For v2f lossless source coding it is required that the parsing dictionary for the $|\mathcal{Y}|$-ary source be complete so that we can parse any right infinite input sequence. Tunstall coding leads to a v2f source encoder for a Discrete Memoryless Source(DMS) P_Y, which has the minimum number of output bits per input symbol among all the v2f source encoders with the same output block length $\lceil \log N \rceil$.. This is because for a v2f encoder output block length and hence the number of

parsing strings in the variable length input dictionary are fixed and Tunstall coding maximizes the expected input length $\mathbb{E}[\ell(T)]$ in such a scenario.

- Define $\mu_Y = \min_{y \in \text{supp}(P_Y)} P_Y(y)$. The extension rule of the algorithm implies

$$\min_{x \in \mathcal{L}(\mathcal{X})} P_Y^{\mathcal{L}(\mathcal{X})}(x) \geq \mu_Y \max_{x \in \mathcal{L}(\mathcal{X})} P_Y^{\mathcal{L}(\mathcal{X})}(x). \tag{2.22}$$

Suppose at some step i the above mentioned property is not satisfied by T_i. Denote by k the branching node that has $t_{i,\min} = \operatorname{argmin}_t P_Y^{\mathcal{L}(\mathcal{T}_i)}(t)$ as successor. Note that k was the leaf node extended in $i-1$ step. Now

$$P_Y^{\mathcal{T}_i}(k) \overset{(a)}{=} P_Y^{\mathcal{T}_{i-1}}(k) \tag{2.23}$$

$$\overset{(b)}{=} \frac{P_Y^{\mathcal{T}_i}(t_{i,\min})}{\mu_Y} \tag{2.24}$$

where (a) is because $\mathcal{T}_i$ is an extension of $\mathcal{T}_{i-1}$ and (2.18) and (b) is because the child with the minimum probability in such a rooted tree with all branching distributions equal to P_Y will have a factor of μ_Y multiplied to its predecessor node probability. Define $j = \operatorname{argmax}_t P_Y^{\mathcal{L}(\mathcal{T}_i)}(t)$. By our assumption we have

$$P_Y^{\mathcal{L}_i}(j) > \frac{P_Y^{\mathcal{L}_i}(t_{i,\min})}{\mu_Y} = P_Y^{\mathcal{T}_i}(k) \tag{2.25}$$

This means j cannot be a child of k since its probability is higher than the probability of the branching node k and it is not possible for a child of k to have higher induced probability than k in a rooted tree with probabilities as long as P_Y is a probability distribution. Hence j was a part of T_{i-1} as well. Moreover $P_Y^{\mathcal{T}_{i-1}}(j) = P_Y^{\mathcal{T}_i}(j)$ using the same argument as earlier for k. This implies

$$P_Y^{\mathcal{T}_{i-1}}(j) > P_Y^{\mathcal{T}_{i-1}}(k) \tag{2.26}$$

But this contradicts the construction rule for the algorithm which states that the leaf with maximum induced leaf probability must be extended at each step hence k must have highest leaf probability. Hence by contradiction we have proved the statement.

For any probability distribution P with $|\operatorname{supp}(P)| = N$ we have

$$1 = \sum_{i \in \operatorname{supp}(P)} P(i) \tag{2.27}$$

$$\geq |\operatorname{supp}(P)| \min_{i \in \operatorname{supp}(P)} P(i) \tag{2.28}$$

$$= N \min_{i \in \operatorname{supp}(P)} P(i) \tag{2.29}$$

This implies

$$\min_{i \in \operatorname{supp}(P)} P(i) \leq \frac{1}{N}. \tag{2.30}$$

Similarly we can show that

$$\max_{i \in \operatorname{supp}(P)} P(i) \geq \frac{1}{N}. \tag{2.31}$$

Since $\mathcal{X}$ is a complete tree $P_Y^{\mathcal{L}(\mathcal{X}}$ is a probability distribution. Combining (2.30) and (2.31) with (2.22) we have

$$\frac{1}{N} \cdot \mu_Y \leq \min_x P_Y^{\mathcal{L}(\mathcal{X})}(x) \leq \frac{1}{N} \tag{2.32}$$

$$\frac{1}{N} \cdot \frac{1}{\mu_Y} \geq \max_x P_Y^{\mathcal{L}(\mathcal{X})}(x) \geq \frac{1}{N} \tag{2.33}$$

From (2.32) and (2.33) we conclude that the entries of the induced leaf distribution $P_Y^{\mathcal{L}(\mathcal{X})}$ of the Tunstall code differ from the entries of the uniform distribution by at most a factor which is independent of N and only depends on P_Y.

2.7.2 Optimal M-type Quantizations of Probability Distributions

In this section we discuss the algorithms to find the M-type probability distribution P'_X that optimally approximates a given distribution P_X such that some measure of resemblance is optimized.

2.7.2.1 Variational Distance Optimal M-type Quantization of a Distribution

For any probability distribution P defined over any finite alphabet $\mathcal{X}$,let P' be the M-type probability distribution that minimizes $||P' - P||_1$, i.e., the variational distance. P' can be found efficiently by [9, Alg. 1]. P'_X has the property

$$|P'(i) - P(i)| \leq \frac{1}{M} \qquad \forall i \in \mathcal{X} \tag{2.34}$$

2.7.2.2 Informational Divergence Optimal M-type Quantization of a Probability Distribution

The M-type distribution P' that minimizes $\mathbb{D}(P'\|P)$ can be found by [9, Alg. 2]. Note that the same algorithm can also be used to optimize informational divegence over all K-type distributions where $1 \leq k \leq M$ as pointed out in [9, Corollary. 1].

Remark 1. *Denote by $P'(M)$ and $P'(KM)$, the informational divergence optimal M-type and KM-type approximations of some distribution P respectively for some positive integers K and M. Then*

$$\mathbb{D}(P'(M)\|P) \geq \mathbb{D}(P'(KM)\|P) \tag{2.35}$$

2.8 References

Definitions and properties in Sec. 2.1 are taken from [10] except Def. 2 which is taken from [3]. The framework in Sec. 2.5 has been adopted from [11] and was extended in [12]. Parts of the discussion on Tunstall coding in Sec. 2.7.1 has been stated in [8] and the methods presented in Sec. 2.7.2 are taken from [9].

3 Problem Statement

Simulation of a DMS P_Y, the "basic problem" dealt with in this thesis, is depicted in the Fig. 3.1. P_B denotes a DMS generating uniformly distributed bits i.e., $P_B = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$. Our aim is to an design encoder which transforms the output of DMS P_B into strings x corresponding to the leaves of a complete $|\mathcal{Y}|$-ary tree $\mathcal{X}$ such that the distribution P_X on $\mathcal{X}$ at the output of the encoder resembles $P_Y^{\mathcal{L}(\mathcal{X})}$. The two important parameters we are interested in for the design of the encoder are rate and distortion. Rate measures the average number of bits required per output symbol by the encoder and distortion measures the resemblance between the approximating distribution P_X and the target distribution $P_Y^{\mathcal{L}(\mathcal{X})}$.
We will deal with two variants of the "basic problem" in this thesis. These are

- Distribution Matching.
- Resolution Coding for Target Distributions.

In the next section we present a generic description of the encoder that we want to design. More specific details will be given in each chapter separately according to the variant of "basic problem" under consideration.

3.1 Encoder

3.1.1 Construction

The encoder we intend to design consists of 3 components

- Complete binary tree $\mathcal{U}$. We will also refer to it as dictionary.
- Complete $|\mathcal{Y}|$-ary tree $\mathcal{X}$. We will also refer to it as codebook.

$$\boxed{P_B} \rightarrow \boxed{\text{Encoder}} \overset{\approx P_Y^{\mathcal{L}(\mathcal{X})}}{\longrightarrow}$$

Figure 3.1: Simulation of DMS P_Y.

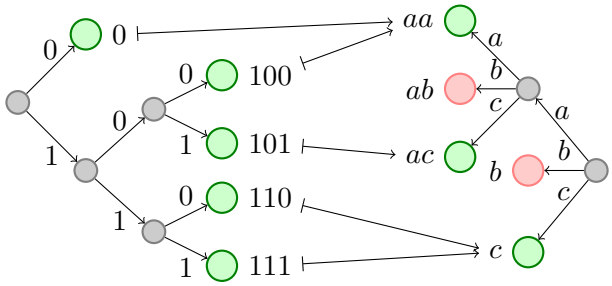

Figure 3.2: An encoder for ternary $\mathcal{Y}$ and a deterministic mapping f, which is many-to-one and not onto. $\mathcal{L}(\mathcal{U}) = \{0, 100, 101, 110, 111\}$ and $\mathcal{L}(\mathcal{X}) = \{aa, ab, ac, b, c\}$

- $f : \mathcal{L}(\mathcal{U}) \to \mathcal{L}(\mathcal{X})$. This mapping can be deterministic or probabilistic, many-to-one , one-to-one,one-to-many (in a probabilistic sense) or a mix of all these. Moreover it is important to note that it is not required that the mapping is onto. For notational convenience we write $f \colon \mathcal{U} \to \mathcal{X}$ instead of $f \colon \mathcal{L}(\mathcal{U}) \to \mathcal{L}(\mathcal{X})$.

We show an example in Fig. 3.2. Encoders discussed in [1, 3, 2] and [6] are all special cases of this encoder construction.

3.1.2 Operation

The encoder starts by parsing a segment of the binary input using the dictionary $\mathcal{U}$. $\mathcal{U}$ needs to be complete so that we can parse any possible right-infinite input sequence. By parsing, we have a DMS $P_U = P_B^{\mathcal{L}(\mathcal{U})}$ with

$$P_U(u) = P_B^{\mathcal{L}(\mathcal{U})}(u) = 2^{-\ell(u)} \qquad \forall \quad u \in \mathcal{L}(\mathcal{U}). \tag{3.1}$$

We have

$$\begin{aligned} \mathbb{H}(P_B^{\mathcal{L}(\mathcal{U})}) &\overset{(a)}{=} \sum_{i \in \mathcal{B}(\mathcal{U})} P_B^{\mathcal{U}}(i)\, \mathbb{H}(P_{B_i}) \\ &\overset{(b)}{=} \sum_{i \in \mathcal{B}(\mathcal{U})} P_B^{\mathcal{U}}(i) \\ &\overset{(c)}{=} \mathbb{E}[\ell(U)] \end{aligned} \tag{3.2}$$

where the expectation in the last equation is w.r.t. $P_B^{\mathcal{L}(\mathcal{U})}$. (a) follows from Leaf Entropy Theorem[13], (b) follows by $P_{B_i} = P_B = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}$ and (c) follows from Path Length Lemma [8].

The encoder then proceeds by generating $X = f(U)$ at the output of the encoder where

X takes values in $\mathcal{L}(\mathcal{X})$. This generates the probability distribution P_X on $\mathcal{L}(\mathcal{X})$. We call P_X the approximating distribution.

3.2 Normalized Informational Divergence and Entropy Rate on Rooted Trees with Probabilities

Let $\mathcal{T}$ be some $|\mathcal{Y}|$-ary complete tree . Let P_T be some distribution defined on $\mathcal{L}$. Denote by $P_Y^{\mathcal{L}}$, the distribution on $\mathcal{L}$ induced by the alphabet distribution P_Y.

Proposition 1. *Entropy Divergence Theorem[12]*

$$\frac{\mathbb{D}(P_T \| P_Y^{\mathcal{L}})}{\mathbb{E}[\ell(T)]} \leq \epsilon \tag{3.3}$$

$$\Rightarrow \left| \frac{\mathbb{H}(P_T)}{\mathbb{E}[\ell(T)]} - \mathbb{H}(P_Y) \right| \leq \delta(\epsilon) \tag{3.4}$$

where the expectation is w.r.t P_T and $\delta(\epsilon) \overset{\epsilon \to 0}{\longrightarrow} 0$.

Proof. See [12] □

This result is important for approximate random number generation. If P_T approximates $P_Y^{\mathcal{L}(\mathcal{T})}$ {exactly, w.r.t. normalized informational divergence or un-normalized informational divergence} then by this implication we have $\frac{\mathbb{H}(P_T)}{\mathbb{E}[\ell(T)]} \to \mathbb{H}(P_Y)$. Entropy rate can be used to establish bounds on the rate of encoders as we will see in the next chapters. These bounds are useful to establish converses. We also use these bounds to show achievability for algorithms proposed in the next chapters.

4 Distribution Matching

Distribution matching is concerned with the *reversible* generation of random numbers, i.e., it should be possible to recover the original bit sequence from the generated symbols with high probability. We start by defining the problem precisely in next section. Then we establish an upper bound on the maximum matching rate and show that we can achieve this upper bound using zero error b2b matchers based on typicality. We then develop more practical zero error variable length matchers and ϵ-error b2b matchers. Finally the relation between distribution matching and lossless source coding is discussed and its application to probabilistic shaping is presented.
A major part of the content discussed in this chapter has been taken from [6, 14] and [15].

4.1 Distribution Matching

In the framework of the "basic problem", we can consider distribution matching as a special case with the following additional restriction.

- f can be a probabilistic mapping, but it is required to be reversible, i.e., with high probability (approaching 1 asymptotically) we can recover the original bit sequence from the output sequence of the encoder, i.e., $\Pr(\hat{U} \neq U) \to 0$ for some $\hat{U} = \varphi(X)$ asymptotically. Note that this concept of reversibility is different from the one in [16] where reversible means that the distortion between P_U and $P_{\hat{U}}$ where $\hat{U} = \varphi(X)$ for some mapping $\varphi : \mathcal{X} \to \mathcal{U}$ (φ can be determisinistic or probabilistic) approaches 0.

In contrast to the "usual" random number generation problem where we are only concerned with the design of an encoder as described in the "basic problem", in the case of distribution matching we also have to discuss the design of a corresponding decoder to ensure the reversibility condition.
The Rate of a matcher is

$$R = \frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \tag{4.1}$$

It is the average number of bits per output symbol. Distortion is characterized by

$$\frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}})}{\mathbb{E}[\ell(X)]} \tag{4.2}$$

i.e., the informational divergence per output symbol.

4.2 Achievable Matching Rates

The matching process can be described as follows. The *encoder* is

$$f \colon \mathcal{U} \to \mathcal{X}, \quad U \mapsto f(U) =: X.$$

The corresponding *decoder* is a mapping

$$\varphi \colon \mathcal{X} \to \mathcal{U}, \quad X \mapsto \varphi(X) =: \hat{U}.$$

Definition 7. *For a given P_Y, we say that a* matching rate R *is* achievable *if there exists a sequence of encoder-decoder pairs that fulfills the following three conditions asymptotically.*

$$\frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \to R \tag{4.3}$$

$$\frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}})}{\mathbb{E}[\ell(X)]} \to 0 \tag{4.4}$$

$$\Pr(U \neq \hat{U}) \to 0. \tag{4.5}$$

4.2.0.1 Maximum Matching Rate

We illustrate by an example that the rate $R \to 0$ can easily be achieved.

Example 1. Let P_Y be a binary distribution $0 < P_Y(0) < 1$.Construct a binary b2b matcher with $\mathcal{U} = \{0, 1\}$, $\mathcal{X} = \{0, 1\}^n$ and the mapping

$$b \mapsto f(b) = b\tilde{Y}^{n-1}, \quad b \in \{0, 1\} \tag{4.6}$$

where $\tilde{Y}^{n-1}$ is distributed according to P_Y^{n-1}. Note that $f(b)$ is a probabilistic mapping.

By the chain rule, the informational divergence is given by

$$\begin{aligned}
&\mathbb{D}(P_X \| P_Y^{\mathcal{C}}) \\
&= \mathbb{D}(P_X \| P_Y^n) \\
&\overset{(a)}{=} \mathbb{D}(P_{\tilde{Y}_1} \| P_Y) + \sum_{y \in \{0,1\}} P_{\tilde{Y}_1}(y)\, \mathbb{D}(P_{\tilde{Y}_2^n | \tilde{Y}_1 = y} \| P_{Y_2^n | Y_1 = y}) \\
&= \mathbb{D}(P_B \| P_Y). \qquad (4.7)
\end{aligned}$$

where we have used the chain rule for informational divergence[12] in (a).Thus, as $n \to \infty$, $\frac{\mathbb{D}(P_X \| P_Y^{\mathcal{C}})}{\mathbb{E}[\ell(X)]} = \frac{\mathbb{D}(P_X \| P_Y^n)}{n} \overset{(a)}{=} \frac{\mathbb{D}(P_B \| P_Y)}{n} \to 0$ where (a) follows by (4.7). Hence (4.4) is satisfied. We define the decoder as

$$\tilde{y}^n \mapsto \varphi_n(\tilde{y}^n) = \tilde{y}_1.$$

Clearly, it decodes with an error probability of zero and (4.5) is fulfilled. Thus, $\{f, \varphi\}$ fulfills our requirements for a matcher with a matching rate of $R = 1/n \to 0$.

This example shows that "small" rates can easily be achieved by using *random* mappings from input symbols to disjoint sets of output symbols. Such mappings allow error free decoding. We are therefore interested in the *maximum* achievable matching rate.

4.3 Converse

We have the following upper bound on the maximum achievable rate for any target distribution P_Y.

Proposition 2. *No rate $R > \mathbb{H}(P_Y)$ is achievable for distribution matching.*

Proof. Estimating U from X: Denote by $\hat{U}$ an estimate of U that results from processing X. Define $P_e := \Pr(U \neq \hat{U})$. Then, by Fano's inequality(Sec. 2.3) we have

$$\begin{aligned}
\mathbb{H}_b(P_e) + P_e \log |\mathcal{U}| &\geq \mathbb{H}(U|\hat{U}) \\
&= \mathbb{H}(U) - [\mathbb{H}(U) - \mathbb{H}(U|\hat{U})] \\
&\overset{(a)}{=} \mathbb{E}[\ell(U)] - \mathbb{I}(U;\hat{U}) \\
&\overset{(b)}{\geq} \mathbb{E}[\ell(U)] - \mathbb{I}(U;X) \\
&= \mathbb{E}[\ell(U)] - \mathbb{H}(X) + \mathbb{H}(X|U) \\
&\overset{(c)}{\geq} \mathbb{E}[\ell(U)] - \mathbb{H}(X)
\end{aligned}$$

where we used (3.2) in (a) and data processing inequality [17, Theo. 1.4] in (b). Note that we can have strict inequality in (c) because $\mathbb{H}(X|U)$ may be nonzero in the case of a random mapping f. Dividing by $\mathbb{E}[\ell(U)]$, we get

$$\begin{aligned}\frac{\mathbb{H}_2(P_e)}{\mathbb{E}[\ell(U)]} + \frac{P_e \log_2(|\mathcal{U}|)}{\mathbb{E}[\ell(U)]} &\geq 1 - \frac{\mathbb{H}(X)}{\mathbb{E}[\ell(U)]}. \\ &= 1 - \frac{\mathbb{H}(X)}{\mathbb{E}[\ell(X)]} \cdot \frac{\mathbb{E}[\ell(X)]}{\mathbb{E}[\ell(U)]}\end{aligned}$$

By Entropy-Divergence theorem (Prop. 1, we know that for $\frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}})}{\mathbb{E}[\ell(X)]} \leq \epsilon$, we have $\mathbb{H}(X)/\mathbb{E}[\ell(X)] \leq \mathbb{H}(P_Y) + \delta(\epsilon)$ such that $\delta(\epsilon) \overset{\epsilon \to 0}{\longrightarrow} 0$. Thus, in the limit,

$$\frac{\mathbb{H}_2(P_e)}{\mathbb{E}[\ell(U)]} + \frac{P_e \log_2(|\mathcal{U}|)}{\mathbb{E}[\ell(U)]} \geq 1 - \frac{\mathbb{E}[\ell(X)]}{\mathbb{E}[\ell(U)]}(\mathbb{H}(P_Y) + \delta(\epsilon)) \tag{4.8}$$

Thus, if the rate $\frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]}$ is larger than $\mathbb{H}(P_Y) + \delta(\epsilon)$, then the probability of error is bounded away from zero. For $\epsilon \to 0$ this requires that $\frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \leq \mathbb{H}(P_Y) + \gamma \quad \forall \gamma > 0$ so that the error probability is not bounded away from zero. This is the statement of the proposition. □

4.4 Zero-Error Achievability

We show that $\mathbb{H}(P_Y)$ is achievable by using a zero error b2b matcher based on typicality. The design of the matcher and the dematcher is as follows. Fix $\epsilon > 0$. Denote by $T_\epsilon^n(P_Y)$ the set of length-n sequences that are ϵ-letter typical with respect to P_Y. From [17, Theorem 4.2], the cardinality of this set is lower bounded by

$$|T_\epsilon^n(P_Y)| \geq [1 - \delta_\epsilon(P_Y, n)] 2^{n(1-\epsilon)\,\mathbb{H}(P_Y)} \tag{4.9}$$

where $\delta_\epsilon(P_Y, n) \overset{n \to \infty}{\longrightarrow} 0$. In particular, there exists an n_0, such that for all $n \geq n_0$, $\delta_\epsilon(P_Y, n) \leq \frac{1}{2}$. Assume $n \geq n_0$. We choose m such that there are 2^m distinct typical sequences:

$$\begin{aligned} m &= \left\lfloor \log_2[1 - \delta_\epsilon(P_Y, n)] \right\rfloor + \left\lfloor n(1-\epsilon)\,\mathbb{H}(P_Y) \right\rfloor \\ &\geq -1 + n(1-\epsilon)\,\mathbb{H}(P_Y) - 1 \\ &= n(1-\epsilon)\,\mathbb{H}(P_Y) - 2. \end{aligned} \tag{4.10}$$

Take $\mathcal{X} = \mathcal{Y}^n$ Let $\mathcal{C} \subseteq T_\epsilon^n(P_Y) \subseteq \mathcal{X}$ be a set of 2^m typical sequences. We define the matcher f_n as a one-to-one mapping from $\{0,1\}^m$ to $\mathcal{C}$ and we define the dematcher as

$\varphi_n = f_n^{-1}$. Note that defining a dematcher by such an inverse mapping is only possible because f is not many to one hence leading to a simple dematcher in principle. We now verify conditions (4.3)–(4.5).

Probability of error: Since the defined mapping is one-to-one, the probability of error is equal to zero for any n.

Informational Divergence: For each sequence $y^n \in \mathcal{C}$, the probability $P_Y^n(y^n)$ is lower bounded [17, Theorem 4.2] by

$$P_Y^n(y^n) \geq 2^{-n(1+\epsilon)\,\mathbb{H}(P_Y)}. \tag{4.11}$$

We calculate

$$\begin{aligned}
\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}) = \mathbb{D}(P_X \| P_Y^n) &= \sum_{y^n \in \mathcal{C}} 2^{-m} \log \frac{2^{-m}}{P_Y^n(y^n)} \\
&\overset{\text{(a)}}{\leq} \sum_{y^n \in \mathcal{C}} 2^{-m} \log \frac{2^{-m}}{2^{-n(1+\epsilon)\,\mathbb{H}(P_Y)}} \\
&= n(1+\epsilon)\,\mathbb{H}(P_Y) - m \\
&\overset{\text{(b)}}{\leq} 2\epsilon n\,\mathbb{H}(P_Y) + 2
\end{aligned} \tag{4.12}$$

where (a) follows from (4.11) and where (b) follows from (4.10). Thus,

$$\begin{aligned}
\lim_{n\to\infty} \frac{\mathbb{D}(P_X \| P_Y^n)}{n} &\leq \lim_{n\to\infty} \frac{2\epsilon n\,\mathbb{H}(P_Y) + 2}{n} \\
&= 2\epsilon\,\mathbb{H}(P_Y).
\end{aligned} \tag{4.13}$$

This holds for any $\epsilon > 0$, which shows that condition (4.4) is fulfilled.

Rate: By (4.10), the rate m/n is bounded as

$$\frac{m}{n} \geq (1-\epsilon)\,\mathbb{H}(P_Y) - \frac{2}{n}. \tag{4.14}$$

Thus, as $\epsilon \to 0$ and $n \to \infty$, $m/n \geq \mathbb{H}(P_Y) - \gamma$ for any $\gamma > 0$.

Remark 2. *Note that $\epsilon \to 0$ drives both the normalized informational divergence in* (4.13) *to zero and the entropy rate in* (4.14) *to $\mathbb{H}(P_Y)$. This exemplifies the relation between informational divergence and entropy that we stated in Prop. 1.*

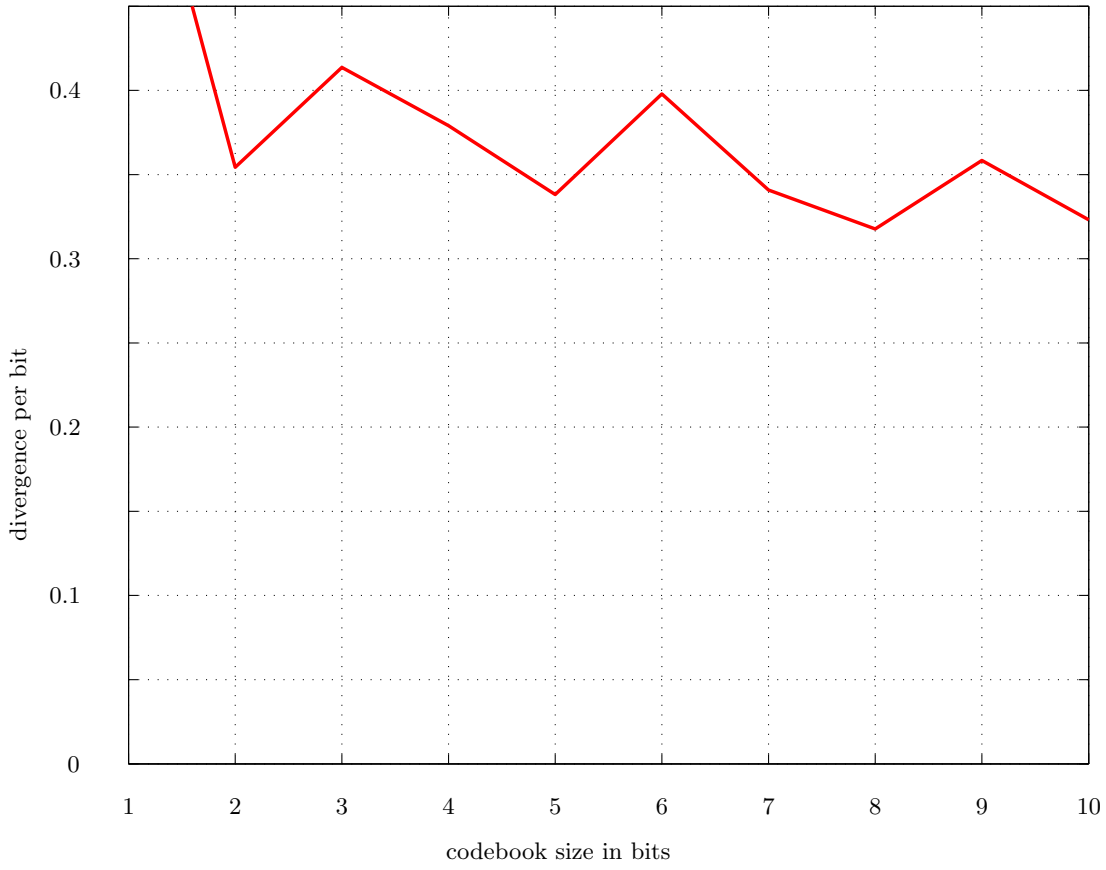

Figure 4.1: Distortion performance of optimal one-to-one b2b matcher w.r.t increasing codebook size m in bits for $P_Y(0) = 1 - P_Y(1) = 0.2145$.

4.4.1 Optimal One-to-One Block-to-Block DistributionMatching

The optimal one-to-one b2b matcher for a fixed input length m chooses the output length n and the codebook $\mathcal{C} \subseteq \{0,1\}^n$ for which the I-divergence per output bit is minimal. To find the optimal one-to-one b2b matcher, we need to solve the optimization problem

$$\underset{n}{\text{minimize}} \left\{ \underset{\mathcal{C} \subseteq \{0,1\}^n \colon |\mathcal{C}| = 2^m}{\text{minimize}} \left\{ \frac{1}{n} \sum_{y^n \in \mathcal{C}} 2^{-m} \log_2 \frac{2^{-m}}{P_Y^n(y^n)} \right\} \right\}.$$

Minimizing over n can be done by a line search around $n \approx m / \mathbb{H}(P_Y)$ and for each n, the codebook $\mathcal{C}$ that minimizes the I-divergence is the one that contains the 2^m sequences from $\{0,1\}^n$ that are most probable according to P_Y^n.

Figure 4.1 shows the distortion performance of the optimal one-to-one b2b matcher with

increasing codebook size, i.e., increasing complexity. Note that this is not the rate-distortion performance of optimal one-to-one b2b matcher. We can see that even for considerable codebook sizes of 2^{10} still we have a large value of distortion and the decay in the distortion with increasing complexity is slow. This motivates us to look if we can somehow improve the distortion vs complexity performance. For optimal one-to-one b2b matchers we have two additional restrictions not required for a distribution matching encoder in general.

- Block-to-block constraint
- One-to-one mapping constraint.

In the next section we will see what can we gain in terms of distortion performance vs complexity when we loosen the b2b constraint but still keeping the one-to-one constraint.

4.5 Variable-to-Fixed Length Distribution Matching

Variable-to-fixed length distribution matching. We will restrict f to be a deterministic one-to-one mapping which implies zero error. The dematcher for one-to-one matching is simple the inverse mapping of f.
For a given ouput length n, first we minimize distortion. Since n is fixed and we only focus on one-to-one matching, minimizing distortion is equivalent to minimizing $\mathbb{D}(P_X\|P_Y^n) = \mathbb{D}(P_B^{\mathcal{U}}\|P_Y^n)$ where the optimization is over $\mathcal{U}$. Later we discuss the rate performance of the v2f matcher which minimizes the distortion for a fixed n.We know from (3.1)

$$P_U(u) = 2^{-\ell(u)} \tag{4.15}$$

i.e. P_U is a dyadic probability distribution. For a deterministic one-to-one mapping f we have $P_X = P_U$. Hence we can only generate dyadic approximating distributions by a v2f encoder with deterministic one-to-one mapping f. Since we can choose $\mathcal{U}$ to be any complete binary dictionary with $|\mathcal{L}(\mathcal{U})| = |\mathcal{Y}|^n$, hence we can generate any dyadic approximating distribution P_X with $|\mathcal{Y}|^n$ entries. Hence to minimize distortion we need to find the informational divergence optimal dyadic approximation P_X of P_Y^n. Constructing $\mathcal{U}$, and hence the v2f matcher since f is one-to-one and $\mathcal{X} = \mathcal{Y}^n$, to generate this approximating distribution is a trivial task since P_X is a dyadic distribution and negative log of the entries of P_X will lead to the desired lengths of $u \in \mathcal{U}$. Hence the problem of designing a v2f matcher with one-to-one deterministic mapping to minimize the distortion

boils down to the following problem

$$\begin{aligned} P_X &= \operatorname*{argmin}_P \mathbb{D}(P\|P_Y^n) \\ &\text{subject to } P \text{ is a dyadic distribution.} \end{aligned} \tag{4.16}$$

Remark 3. *Having the constraint that a dyadic distribution P has K elements implies that $P(i) \geq 2^{-K+1} \quad \forall i$.*

4.5.1 Geometric Huffman Coding

Geometric Huffman Coding (GHC) is an algorithm to solve the problem of finding the informational divergence optimal dyadic approximation of any distribution P. It was proposed in [18]. For $P = P_Y^n$ it solves the problem stated in (4.16). It follows a procedure similar to Huffman Coding [19]. The updating rules are as follows. Without loss of generality assume $P(1) \geq P(2) \geq \cdots \geq P(q)$. The first rule dictates how this problem with q entries can be reduced to a problem with $q-1$ entries and the second rule states how the prefix free tree for this is constructed.

- The two smallest entries $P(q)$ and $P(q-1)$ are replaced by $P'(q-1)$ with the following rule.

$$P'(q-1) = \begin{cases} P(q-1) & \text{if} \quad P(q-1) \geq 4P(q) \\ 2\sqrt{P(q-1)P(q)} & \text{if} \quad P(q-1) < 4P(q) \end{cases} \tag{4.17}$$

- We use the following rule to update the tree at each step.
  - if $P(q-1) \geq 4P(q)$: Remove the whole subtree that originates from node q and associate probability 0 to it.
  - if $P(q-1) < 4P(q)$: join node q and node $q-1$ in a parent node.

The complexities of GHC and Huffman coding are the amei.e. $\mathcal{O}(q \log q)$. The optimality of GHC is shown in [6, Sec. 3.2].

4.5.2 Distortion

In [6] it has been shown that for $n \to \infty$, $\frac{\mathbb{D}(P_X\|P_Y^n)}{n} \to 0$ where $P_X = \mathrm{GHC}(P_Y^n)$.

4.5.3 Rate

For v2f matching

$$R = \frac{\mathbb{E}[\ell(U)]}{n}$$

We know from (3.1)

$$\mathbb{E}[\ell(U)] = \mathbb{H}(P_U)$$

and for deterministic one-to-one mapping f we have $P_U = P_X$ hence

$$R = \frac{\mathbb{H}(P_X)}{n}. \tag{4.18}$$

For $\frac{\mathbb{D}(P_X \| P_Y^n)}{n} \to 0$ we know that $\frac{\mathbb{H}(P_X)}{n} \to \mathbb{H}(P_Y)$ from Prop. 1. Hence using the result in Sec. 4.5.2, as $n \to \infty$ we have

$$R \to \mathbb{H}(P_Y). \tag{4.19}$$

Hence this v2f matcher achieves the upper bound $\mathbb{H}(P_Y)$ asymptotically.

4.6 Fixed-to-Variable Length Distribution Matching

For f2v distribution matching we have $\mathcal{U} = \{0,1\}^m$ and hence $P_U(u) = 2^{-m}, \quad \forall u \in \mathcal{U}$. Again we will only focus on distribution matching using deterministic one-to-one mapping for the same reason as mentioned for v2f matching. Fixed length input and one-to-one mapping leads to $P_X = P_U$ i.e the approximating distribution is uniform. Our goal in the design of the matcher is minimizing the distortion for a given m. Moreover, unlike v2f matching, we will only discuss f2v matching for binary target distribution P_Y. This is done to keep the discussion more tractable and more importantly to show the interesting relationship that exists between binary f2v distribution matching and Tunstall coding for binary sources. This relationship is not present in case of non-binary f2v distribution matching. For non binary f2v distribution matching we can establish a similar relation to Varn's algorithm [20, Problem. C2] but we will not delve into the details of this relation in this thesis.

Our aim is to design f2v distribution matcher with one-to-one mapping for binary target distribution P_Y such that it minimizes the distortion.This can be expressed as solving the optimization problem:

$$\min_{\mathcal{X}, f:\mathcal{U}\to\mathcal{X}'} \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} \tag{4.20}$$

where $\mathcal{X}$ can be any $|\mathcal{Y}|$-ary complete tree with $|\mathcal{L}(\mathcal{X})| \geq 2^m$ and $\mathcal{L}(\mathcal{X}') \subseteq \mathcal{L}(\mathcal{X})$ such that $|\mathcal{L}(\mathcal{X}')| = 2^m$ hence f is onto $\mathcal{X}'$. Basically $\mathcal{L}(\mathcal{X}')$ denotes the subset of $\mathcal{L}(\mathcal{X})$ such that $\mathcal{L}(\mathcal{X}')$ is the image of f. For a one-to-one mapping f the only freedom we have for f is to choose the subset of $\mathcal{X}$ we choose to be the image of f. The actual assignment $X = f(U)$ is not important as long as the set $\mathcal{L}(\mathcal{X}')$ is unchanged since $P_U(u)$ is a uniform random variable hence reordering doesnot change the approximating distribution P_X. Note that

$\mathcal{X}'$ may not be a complete tree. Futhermore, Note that choosing $x \in \mathcal{L}(\mathcal{X})$ (corresponding to minimizing $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})$) as we did in Sec. 4.4.1 for b2b matching may not be optimal since choosing these most probable $x \in \mathcal{L}(\mathcal{X})$ may lead to a smaller value of $\mathbb{E}[\ell(X)]$ and the overall normalized informational divergence may be higher than if we had chosen some other set of $x \in \mathcal{L}(\mathcal{X})$. To solve the problem in (4.20) we proceed as follows. First we will show that Tunstall coding is optimal for minimizing $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})$. Then we will construct an iterative procedure using Tunstall coding to minimize $\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}_{P_X}[L(X)]}$ when $\mathcal{X}'$ is restricted to be $\mathcal{X}$ i.e. we only allow $\mathcal{X}$ with 2^m leaves. Subsequently we derive the conditions under which the optimal solution for normalized divergence has the property that $\mathcal{X}' = \mathcal{X}$ and hence our iterative procedure is optimal.

4.6.1 Minimizing Informational divergence

Definition 8. *Let* $\mathbf{R}$ *be the set of real numbers. For a finite set* $\mathcal{Y}$*, we say that* $W \colon \mathcal{Y} \to \mathbf{R}$ *is a* weighted distribution *if for each* $i \in \mathcal{Y}$*,* $W(i) > 0$*. We allow for* $\sum_{i \in \mathcal{Y}} W(i) \neq 1$*.*

The I-divergence of a distribution P and a weighted distribution W is

$$\mathbb{D}(P \| W) = \sum_{i \in \operatorname{supp} P} P(i) \log_2 \frac{P(i)}{W(i)} \tag{4.21}$$

The reason why we need this generalization of the notion of distributions and informational divergence will become clear in the next section.

Proposition 3. *Let* P_Y *be a weighted binary target distribution. Define* $\mathcal{T}^*$

$$\begin{aligned} \mathcal{T}^* = \operatorname*{argmin}_{\mathcal{T}} \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) \\ \text{Subject to } \mathcal{T} \text{ is a binary complete dictionary and } |\mathcal{L}(\mathcal{T})| = 2^m. \end{aligned} \tag{4.22}$$

Hence $f \colon \mathcal{U} \to \mathcal{T}$ *is onto. Then we find that*

i. An optimal $\mathcal{T}^*$ *can be constructed by applying Tunstall coding to* P_Y*.*

ii. If $0 \leq P_Y(0) \leq 1$ *and* $0 \leq P_Y(1) \leq 1$*, then* $\mathcal{T}^*$ *also minimizes* $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})$ *among all possible* $\mathcal{T}$ *s.t.* $|\mathcal{L}(\mathcal{T})| \geq 2^m$ *and* $f : \mathcal{U} \to \mathcal{T}$ *is possibly not onto.*

Proof. Part i. We write

$$\begin{aligned}\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{T})}) &= \sum_{i\in\mathcal{L}} 2^{-m}\log\frac{2^{-m}}{P_Y^{\mathcal{L}}(i)}\\ &= -m - 2^{-m}\sum_{i\in\mathcal{L}}\log P_Y^{\mathcal{L}}(i)\end{aligned} \tag{4.23}$$

and hence

$$\operatorname*{argmin}_{\mathcal{T}}\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{T})}) = \operatorname*{argmax}_{\mathcal{T}}\sum_{i\in\mathcal{L}(\mathcal{T})}\log P_Y^{\mathcal{L}(\mathcal{T})}(i). \tag{4.24}$$

Note that the summation on the R.H.S is over all leaves of $\mathcal{T}$ since f is onto in this case. Consider now an arbitrary tree $\mathcal{T}$ s.t. $|\mathcal{L}(\mathcal{T})| = 2^m$. Since $\mathcal{T}$ is complete by definition, there exist (at least) two leaves that are siblings, say j and $j+1$. Denote by k the corresponding parent node. The contribution of these two leaves to the objective function on the R.H.S. of (4.24) can be written as

$$\begin{aligned}\log P_Y^{\mathcal{L}(\mathcal{T})}(j) + \log P_Y^{\mathcal{L}(\mathcal{T})}(j+1) &= \log[P_Y^{\mathcal{L}(\mathcal{T})}(k)P_Y(0)] + \log[P_Y^{\mathcal{L}(\mathcal{T})}(k)P_Y(1)]\\ &= \log P_Y^{\mathcal{L}(\mathcal{T})}(k) + \log P_Y^{\mathcal{L}(\mathcal{T})}(k) + \log P_Y(0) + \log P_Y(1).\end{aligned} \tag{4.25}$$

Now consider the tree $\mathcal{T}'$ that results from removing the nodes j and $j+1$. The new set of leaves is $\mathcal{L}(\mathcal{T}') = k \cup \mathcal{L}(\mathcal{T}) \setminus \{j, j+1\}$ and the new set of branching nodes is $\mathcal{B}(\mathcal{T}') = \mathcal{B}(\mathcal{T}) \setminus k$. Also $P_Y^{\mathcal{L}(\mathcal{T}')}$ defines a weighted leaf distribution on $\mathcal{L}(\mathcal{T}')$ and $|\mathcal{L}(\mathcal{T}')| = 2^m - 1$. The same procedure can be applied repeatedly by defining $\mathcal{T} = \mathcal{T}'$, until $\mathcal{T}'$ consists only of the root node. We use this idea to re-write the objective function of the R.H.S. of (4.24) as follows.

$$\begin{aligned}&\sum_{i\in\mathcal{L}(\mathcal{T})}\log P_Y^{\mathcal{L}(\mathcal{T})}(i)\\ &\quad= \sum_{i\in\mathcal{L}(\mathcal{T}')}\log P_Y^{\mathcal{L}(\mathcal{T}')}(i) + \log P_Y^{\mathcal{T}}(k) + \log P_Y(0) + \log P_Y(1)\\ &\quad= \sum_{k\in\mathcal{B}(\mathcal{T})}\log P_Y^{\mathcal{T}}(k) + (2^m-1)[\log P_Y(0) + \log P_Y(1)].\end{aligned} \tag{4.26}$$

We have the factor $2^m - 1$ since for the original tree $|\mathcal{L}(\mathcal{T})| = 2^m$ and by definition $\mathcal{T}$ is complete. For any binary complete tree $\mathcal{T}$, the number of branching nodes are —$\mathcal{L}(\mathcal{T})$ —-1. At every step the new $\mathcal{T}$ has one leaf less and an added term of $\log P_Y(0) + \log P_Y(1)$, hence ultimately leading to the factor $2^m - 1$. Since $(2^m - 1)[\log P_Y(0) + \log P_Y(1)]$ is a constant independent of the structure of the complete tree $\mathcal{T}$ and only

depends on $|\mathcal{L}(\mathcal{T})|$, we have

$$\begin{aligned} &\underset{\mathcal{T}}{\operatorname{argmax}} \sum_{i\in\mathcal{L}(\mathcal{T})} \log P_Y^{\mathcal{L}(\mathcal{T})}(i) = \underset{\mathcal{T}}{\operatorname{argmax}} \sum_{k\in\mathcal{B}(\mathcal{T})} \log P_Y^{\mathcal{T}}(k). \\ &\text{Subject to } \mathcal{T} \text{ is a complete tree and } |\mathcal{L}(\mathcal{T})| = 2^m \end{aligned} \tag{4.27}$$

The R.H.S. of (4.27) is clearly maximized by the complete tree with the branching nodes with the greatest weighted probabilities. According to [8, p. 47] and the discussion in Sec. 2.7.1, this is exactly the tree that is constructed when Tunstall coding is applied to the weighted distribution P_Y.

Part ii. We now consider $P_Y(0) \leq 1$ and $P_Y(1) \leq 1$. Assume we have $|\mathcal{L}(\mathcal{T})| > 2^m$ and $f : \mathcal{U} \to \mathcal{T}_1$ s.t $\mathcal{L}(\mathcal{T}_1) \subset \mathcal{L}(\mathcal{T})$ and $|\mathcal{L}(\mathcal{T}_1)| = 2^m$. Since f is not an onto mapping on $\mathcal{L}(\mathcal{T})$ (but is an onto mapping on $\mathcal{L}(\mathcal{T}_1)$ which corresponds to an incomplete tree) we can find a branching node j such that all the leaves of $\mathcal{L}(\mathcal{T})$ in the subtree emerging from one of the two immediate children of j is such that no leaf is element of $\mathcal{L}(\mathcal{T}_1)$. Without loss of generality, assume that this branch is labeled by a one. Remove this branch. Denote by $\mathcal{S}$ the set of leaves on the subtree of the other branch. For any leaf i in $\mathcal{S}$ the new probability is $\frac{P_Y^{\mathcal{L}(\mathcal{T})}(i)}{P_Y(0)}$. Using the same mapping f as earlier i.e. the same set of leaves but now the labels of the leaves emerging from $\mathcal{S}$ have changed and have specifically one "zero" missing. Thus, for the new complete tree $\mathcal{T}'$, the objective function (4.26) is bounded as

$$\begin{aligned} \sum_{i\in\mathcal{L}(\mathcal{T}')} \log P_Y^{\mathcal{L}(\mathcal{T}')}(i) &= \sum_{i\in\mathcal{L}(\mathcal{T})\setminus S} \log P_Y^{\mathcal{L}(\mathcal{T})}(i) + \sum_{i\in\mathcal{S}} \log \frac{P_Y^{\mathcal{L}(\mathcal{T})}(i)}{P_Y(0)} \\ &\geq \sum_{i\in\mathcal{L}(\mathcal{T})} \log P_Y^{\mathcal{L}(\mathcal{T})}(i). \end{aligned} \tag{4.28}$$

In summary, under the assumption $P_Y(0) \leq 1$ and $P_Y(1) \leq 1$, for every $\mathcal{T}$ such that $|\mathcal{L}(\mathcal{T})| > 2^m$ we can find $\mathcal{T}'$ s.t. $|\mathcal{L}(\mathcal{T}')| < |\mathcal{L}(\mathcal{T})|$ and the objective function (4.26) is improved . This proves the statement ii. of the proposition. □

4.6.2 Minimizing informational Divergence Per Bit

The following two propositions relate the problem of minimizing the informational divergence per output bit (normalized informational divergence) to the problem of minimizing the un-normalized informational divergence.

Let $\mathbb{T}$ be some set of complete binary trees with at least 2^m leaves (note that for more

than 2^m leaves f will not be onto). Define

$$\Delta := \min_{\mathcal{T}\in\mathbb{T}} \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]}. \tag{4.29}$$

Remember that $P_T = P_U$ is the uniform distribution with cardinality 2^m.

Proposition 4. *We have*

$$\mathcal{T}^* := \operatorname*{argmin}_{\mathcal{T}\in\mathbb{T}} \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} = \operatorname*{argmin}_{\mathcal{T}\in\mathbb{T}} \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}),\Delta}) \tag{4.30}$$

where $P_Y^{\mathcal{L}(\mathcal{T}),\Delta}$ is the weighted distribution induced on $\mathcal{L}(\mathcal{T})$ by the weighted distribution $P_Y \circ 2^\Delta := [P_Y(0)2^\Delta, P_Y(1)2^\Delta]$ on $\mathcal{Y} = \{0,1\}$.

Proof. Denote by $\mathbb{T}'$ some set of complete trees with greater than or equal to 2^m leaves, By (4.29), for any tree $\mathcal{T} \in \mathbb{T}'$, we have

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} \geq \Delta \text{ with equality if } \mathcal{T} = \mathcal{T}^* \tag{4.31}$$

$$\Rightarrow \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) - \Delta\,\mathbb{E}[\ell(T)] \geq 0$$
$$\text{with equality if } \mathcal{T} = \mathcal{T}^*. \tag{4.32}$$

We write the L.H.S. of (4.32) as

$$\begin{aligned}
&\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) - \Delta\,\mathbb{E}[\ell(T)] \\
&\qquad = \sum_{i\in\mathrm{supp}(P_U)} P_U(i) \log \frac{P_U(i)}{P_Y^{\mathcal{L}(\mathcal{T})}(i)} - \Delta \sum_{i\in\mathcal{L}} P_U(i)\ell(i) \\
&\qquad = \sum_{i\in\mathrm{supp}(P_U)} P_U(i) \left[\log \frac{P_U(i)}{P_Y^{\mathcal{L}(\mathcal{T})}(i)} - \log 2^{\Delta\ell(i)}\right] \\
&\qquad = \sum_{i\in\mathrm{supp}(P_U)} P_U(i) \log \frac{P_U(i)}{P_Y^{\mathcal{L}(\mathcal{T})}(i)2^{\Delta\ell(i)}}.
\end{aligned} \tag{4.33}$$

Consider the path through the tree that ends at leaf i. Denote by ℓ_i^0 and ℓ_i^1 the number of times the labels 0 and 1 occur, respectively. The length of the path is $\ell(i) = \ell_i^0 + \ell_i^1$.

The term $P_Y^{\mathcal{L}(\mathcal{T})}(i)2^{\Delta\ell_i}$ can now be written as

$$\begin{aligned} P_Y^{\mathcal{L}(\mathcal{T})}(i)2^{\Delta\ell(i)} &= P_Y^{\ell_i^0}(0)P_Y^{\ell_i^1}(1)2^{\Delta(\ell_i^0+\ell_i^1)} \\ &= (P_Y(0)2^{\Delta})^{\ell_i^0}(P_Y(1)2^{\Delta})^{\ell_i^1} \\ &= P_Y^{\mathcal{L}(\mathcal{T}),\Delta}(i). \end{aligned} \tag{4.34}$$

Using (4.34) and (4.33) in (4.32) shows that for any binary tree $\mathcal{T} \in \mathbb{T}$ we have

$$\sum_{i\in\mathrm{supp}(P_U)} P_U(i)\log_2\frac{P_U(i)}{P_Y^{\mathcal{L}(\mathcal{T}),\Delta}(i)} \geq 0 \text{ with equality if } \mathcal{T}=\mathcal{T}^* \tag{4.35}$$

which is the statement of the proposition. □

Remark 4. *The same technique of converting a problem of minimizing normalized objective function into an un-normalized one has been used in [6, Chap. 4].*

Remark 5. *Note that Prop. 4 is also valid if P_T is not uniform.*

Proposition 5. *Define*

$$\Delta_c := \min_{\mathcal{T}} \frac{\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]}. \tag{4.36}$$

where the optimization is over all complete *trees $\mathcal{T}$ such that $|\mathcal{L}(\mathcal{T})| = 2^m$*

$$\mathcal{T}^* := \operatorname*{argmin}_{\mathcal{T}} \frac{\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} \tag{4.37}$$

is constructed by applying Tunstall coding to $P_Y \circ 2^{\Delta}$.

Proof. The proposition is a consequence of Prop. 4 and Prop. 3.i. □

4.6.3 Iterative Algorithm

By Prop. 5, if we know the informational divergence Δ_c, then we can find $\mathcal{T}^*$ by Tunstall coding. For our matcher design, $\mathcal{X} = \mathcal{T}^*$ and since $|\mathcal{L}(\mathcal{T}^*)| = 2^m$, f is an onto mapping on T^*. However, Δ_c is not known a priori. We solve this problem by iteratively applying Tunstall coding to $P_Y \circ 2^{\hat{\Delta}}$, where $\hat{\Delta}$ is an estimate of Δ_c and by updating our estimate. This procedure is stated in Alg. 1. The stopping criterion mentioned in Alg. 1 can also be restated as $\hat{\Delta}_i = \hat{\Delta}_{i+1}$. It should be noted that in this algorithm we are only doing optimization over $\mathcal{T}$ such that $|\mathcal{L}(\mathcal{T})| = 2^m$ as stated in Prop. 5

Algorithm 1.

$\hat{\mathcal{T}} \leftarrow \underset{\mathcal{T}\text{s.t.}|\mathcal{L}(\mathcal{T})|=2^m}{\operatorname{argmin}} \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})$ *solved by Tunstall coding on* P_Y

repeat

1. $\hat{\Delta} = \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}})})}{\mathbb{E}[L(\hat{\mathcal{T}})]}$
2. $\hat{\mathcal{T}} = \underset{\mathcal{T}\text{s.t.}|\mathcal{L}(\mathcal{T})|=2^m}{\operatorname{argmin}} \left[\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) - \hat{\Delta}\, \mathbb{E}[L(\mathcal{T})]\right]$ *Tunstall on* $P_Y \circ 2^{\hat{\Delta}}$

until $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}})}) - \hat{\Delta}\, \mathbb{E}[L(\hat{\mathcal{T}})] = 0$

$\Delta_c = \hat{\Delta}$, $\mathcal{T}^* = \hat{\mathcal{T}}$

Proposition 6. *Alg. 1 finds* $(\Delta, \mathcal{T}^*)$ *as defined in Prop. 5 in finitely many steps.*

Proof. The proof is similar to the proof of [6, Prop. 4.1]. We first show that Δ *is strictly monotonically decreasing.* Let $\hat{\Delta}_i$ be the value that is assigned to $\hat{\Delta}$ in step 1. of the ith iteration and denote by $\hat{\mathcal{T}}_i$ the value that is assigned to $\hat{\mathcal{T}}$ in step 2. of the ith iteration. Suppose that the algorithm does not terminate in the ith iteration. We have

$$\begin{aligned}\hat{\Delta}_i &= \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}}_{i-1})})}{\mathbb{E}[L(\hat{T}_{i-1})]} \\ &\Rightarrow \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}}_{i-1})}) - \hat{\Delta}_i\, \mathbb{E}[L(\hat{T}_{i-1})] = 0.\end{aligned}$$

By step 2, we have

$$\hat{\mathcal{T}}_i = \underset{\mathcal{T}}{\operatorname{argmin}} \left[\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) - \hat{\Delta}_i\, \mathbb{E}[\ell(T)]\right]$$

and since by our assumption the algorithm does not terminate in the ith iteration, we have

$$\begin{aligned}\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}}_i)}) - \hat{\Delta}_i\, \mathbb{E}[L(\hat{T}_i)] < 0 \\ \Rightarrow \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\hat{\mathcal{T}}_i)})}{\mathbb{E}[L(\hat{T}_i)]} < \hat{\Delta}_i \\ \Rightarrow \hat{\Delta}_{i+1} < \hat{\Delta}_i.\end{aligned}$$

Now assume the algorithm terminated, and let $\hat{\mathcal{T}}$ be the tree after termination. Because of the assignments in steps 1. and 2., the terminating condition implies that for any complete tree $\mathcal{T}$ such that $|\mathcal{L}(\mathcal{T})| = 2^m$, we have

$$\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})}) - \hat{\Delta}\, \mathbb{E}[\ell(T)] \geq 0, \text{with equality if } \mathcal{T} = \hat{\mathcal{T}}. \tag{4.38}$$

Consequently, we have

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} \geq \hat{\Delta}, \text{with equality if } \mathcal{T} = \hat{\mathcal{T}}.$$

We conclude that after termination, $(\Delta, \hat{\mathcal{T}})$ is equal to the optimal tuple $(\Delta, \mathcal{T}^*)$ in Prop. 5.

Finally, we have shown that Δ is strictly monotonically decreasing so that $\hat{\mathcal{T}}_i \neq \hat{\mathcal{T}}_j$ for all $i < j$. There is only a finite number of complete binary trees with 2^m leaves. Thus, the algorithm terminates after finitely many steps. □

4.6.4 Optimality of Complete Codebook with 2^m leaves

In the previous section we have presented the iterative algorithm to minimize normalized divergence over $\mathcal{X}$ s.t. $\mathcal{X}$ is complete and $|\mathcal{L}(\mathcal{X})| = 2^m$ for f2v distribution matching.

In general using $\mathcal{X}$ s.t. $|\mathcal{L}(\mathcal{X})| = 2^m$ and hence f is onto is not optimal. We illustrate this by an example.

Example 2. Consider $m = 1$ and P_Y : $P_Y(0) = \frac{5}{6}, P_Y(1) = \frac{1}{6}$. For $m = 1$, Tunstall coding constructs the (unique) complete binary tree $\mathcal{X}$ with 2 leaves, independent of which target vector we pass to it. The path lengths are $\ell(1) = \ell(2) = 1$. The informational divergence per bit achieved by this is

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} = \frac{-1 - \frac{1}{2}\log_2(P_Y(0)P_Y(1))}{1} = 0.424 \text{ bits}.$$

Now, we could instead use a complete tree $\mathcal{T}$ with the leaves $\{0, 10, 11\}$ and $f : \{0, 1\} \to \{0, 10\}$. Note that in this case $|\mathcal{L}(\mathcal{T})| = 3 > 2^m$. In this case, the informational divergence per bit is

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} = \frac{-1 - \frac{1}{2}\log_2(P_Y(0)P_Y(1)P_Y(0))}{\frac{1}{2}(1+2)} = 0.37034 \text{ bits}.$$

In summary, for the considered example, using a complete tree with 2^m leaves is sub-optimal. Additionally, note that $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})}) \leq \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})$ as we saw earlier that un-normalized informational divergence is minimized by Tunstall coding.

We will derive simple conditions on the target vector P_Y in the following subsection that guarantee that the optimal complete tree has 2^m leaves.

4.6.4.1 Sufficient Conditions for Optimality

Proposition 7. *Let P_Y be a distribution. If $\max\{P_Y(0), P_Y(1)\} \leq 4\min\{P_Y(0), P_Y(1)\}$, then the optimal tree among all complete trees $\mathcal{X}$ s.t. $|\mathcal{L}(\mathcal{X})| \geq 2^m$ has 2^m leaves for any $m \geq 1$ and it is constructed by Alg. 1.*

Proof. According to Prop. 3.ii, the tree that minimizes $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}),\Delta})$ is complete if the entries of the weighted distribution $P_Y \circ 2^{\Delta}$ are both less than or equal to one. Without loss of generality, assume that $P_Y(0) \geq P_Y(1)$. Thus, we only need to check this condition for $P_Y(0)$. We have

$$\begin{aligned} P_Y(0) \cdot 2^{\Delta} &\leq 1 \\ \Leftrightarrow \log P_Y(0) + \Delta &\leq 0 \\ \Leftrightarrow \Delta &\leq -\log P_Y(0). \end{aligned} \tag{4.39}$$

We calculate the value of Δ that is achieved by the (unique) complete tree with 2 leaves, namely

$$\Delta = \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T})})}{\mathbb{E}[\ell(T)]} = -1 - \frac{1}{2}\log(P_Y(0)P_Y(1)). \tag{4.40}$$

For each $m \geq 1$, this Δ is achieved by $\mathcal{L}(\mathcal{T}) = \{0,1\}^m$. Substituting the right-hand side of (4.40) for Δ in (4.39), we obtain

$$\begin{aligned} -1 - \frac{1}{2}\log(P_Y(0)P_Y(1)) &\leq -\log P_Y(0) \\ \Leftrightarrow 1 + \log\left[(P_Y(0)P_Y(1))^{\frac{1}{2}}\right] &\geq \log P_Y(0) \\ \Leftrightarrow 2\sqrt{P_Y(0)P_Y(1)} &\geq P_Y(0) \\ \Leftrightarrow 4P_Y(1) &\geq P_Y(0) \end{aligned} \tag{4.41}$$

which is the condition stated in the proposition. □

4.6.4.2 Asymptotic Achievability for Complete Trees with 2^m leaves

Proposition 8. *Denote by $\mathcal{T}(m)$ the complete tree with 2^m leaves that is constructed by applying Alg. 1 to a target distribution P_Y. Then we have*

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{T}(m)})}{\mathbb{E}[L(T(m))]} \leq \frac{\log \frac{1}{\min\{P_Y(0), P_Y(1)\}}}{m} \tag{4.42}$$

and in particular, the informational divergence per bit approaches zero as $m \to \infty$.

Proof. The expected length can be bounded by the converse of the Coding Theorem for DMSs [8, p. 45] as

$$\begin{aligned}\mathbb{E}[L(T(m))] &\geq \mathbb{H}(P_U) \\ &= \mathbb{H}(P_X) \\ &= m. \end{aligned} \tag{4.43}$$

Thus, we have

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}(m))})}{\mathbb{E}[L(T(m))]} \leq \frac{\min_{\mathcal{T}'(m)} \mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}'(m))})}{m}. \tag{4.44}$$

where the optimization on the R.H.S is over all complete trees with 2^m leaves. The tree $\mathcal{T}''(m)$ that minimizes the R.H.S. is found by applying Tunstall coding to P_Y as shown in Prop. 3. Without loss of generality, assume that $P_Y(0) \geq P_Y(1)$.

We can bound the informational divergence as

$$\begin{aligned}\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}''(m))}) &= \sum_{i \in \mathcal{L}(\mathcal{T}''(m))} 2^{-m} \log_2 \frac{2^{-m}}{P_Y^{\mathcal{L}(\mathcal{T}''(m))}(i)} \\ &\overset{a}{\leq} \sum_{i \in \mathcal{L}(\mathcal{T}''(m))} 2^{-m} \log_2 \frac{2^{-m}}{2^{-m} P_Y(1)} \\ &= \log_2 \frac{1}{P_Y(1)}. \end{aligned} \tag{4.45}$$

where we have used (2.32) in (a). We can now bound the informational divergence per bit as

$$\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}(m))})}{\mathbb{E}[L(T(m))]} \leq \frac{\log \frac{1}{P_Y(1)}}{\mathbb{E}[L(T(m))]} \leq \frac{\log \frac{1}{P_Y(1)}}{m}. \tag{4.46}$$

This proves the proposition. □

4.6.4.3 Optimality of Complete Trees with 2^m leaves for Large Enough m

Proposition 9. *For any target distribution P_Y with $P_Y(0) < 1$ and $P_Y(1) = 1 - P_Y(0)$, there is an m_0 such that for all $m > m_0$, the complete tree that minimzes*

$$\begin{aligned} &\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{T}(m))})}{\mathbb{E}[L(T(m))]} \\ &\textit{Subject to } |\mathcal{L}(\mathcal{T})| \geq 2^m \end{aligned} \tag{4.47}$$

has 2^m leaves.

Proof. Without loss of generality, assume that $P_Y(0) \geq P_Y(1)$. By Prop. 8, we have $\Delta_m \leq \frac{\log_2 \frac{1}{P_Y(1)}}{m}$. Thus, there exists an m_0 such that

$$P_Y(1)2^{\Delta_m} \leq P_Y(0)2^{\Delta_m} \leq q_0 2^{\frac{\log_2 \frac{1}{q_1}}{m}} \leq 1, \text{ for all } m > m_0. \tag{4.48}$$

Thus, for all $m \geq m_0$, both entries of $P_Y \circ \Delta_m$ are smaller than 1. The proposition now follows by Prop. 4 and Prop. 3.ii. □

4.6.5 Rate

For f2v distribution matching, the rate R is

$$\begin{aligned} R &= \frac{m}{\mathbb{E}[\ell(X)]} \\ &= \frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \end{aligned} \tag{4.49}$$

Moreover

$$\mathbb{H}(P_X) \overset{(b)}{=} \mathbb{H}(P_U) \overset{(a)}{=} m$$

where (a) is because of fixed length input and (b) is because f is one-to-one.

Hence

$$R = \frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} = \frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} \tag{4.50}$$

Using Divergence Entropy Theorem (Prop. 1) and 8, for Alg. 1 we have as $m \to \infty$,

$$R > \mathbb{H}(P_Y) - \gamma \tag{4.51}$$

for any $\gamma > 0$.

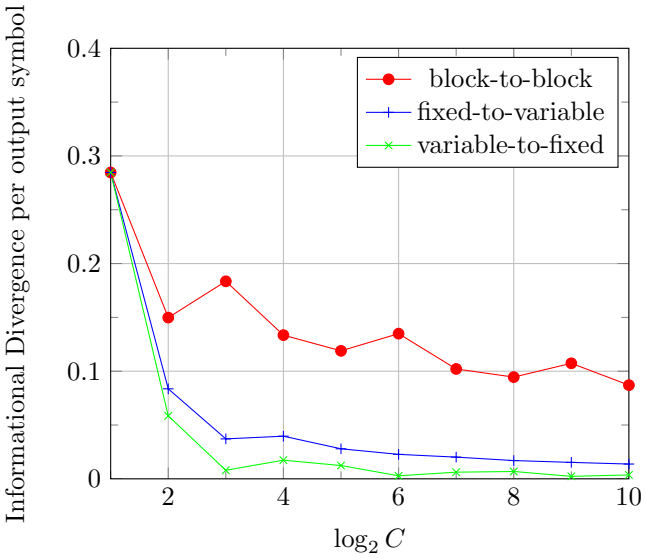

Figure 4.2: Comparison of complexity vs distortion performance of zero error Distribution Matching techniques for target distribution $P_Y(0) = 1 - P_Y(1) = 0.2145$. C represents the codebook size.

4.7 Performance Comparison of Zero-error Matching Techniques

In Fig. 4.2 we show the informational divergence per bit achieved by the three zero error distibution matching techniques discussed in the previous sections w.r.t. the codebook size in bits, i.e., m. We see that using a variable length zero error distribution matcher has a significant gain in terms of distortion vs complexity performance.

4.8 ϵ-Error Block-to-Block Distribution Matching

Although zero error variable length matchers significantly outperform optimal one-to-one b2b matchers, they have three inherent problems, namely *synchronization*, *error propagation*, and *variable transmission rate*. We illustrate this by an example. Consider the v2f matcher

$$1 \mapsto a, 00 \mapsto b, 01 \mapsto c$$

which generates the channel input symbols $\{a, b, c\}$ with probabilities $1/2, 1/4, 1/4$, respectively. The binary string 01001 is mapped to *cba*, which is then transmitted over a noisy channel. The string *aba* is detected at the receiver and according to the matcher

mapped to 1001, i.e.,

$$01001 \mapsto cba$$
$$aba \mapsto 1001.$$

First, the input length is 5 but the output length is 4, so input and output are out of sync. Second, one detection error led to 3 bit errors and one bit is missing. Third, an all b string on the channel corresponds to twice as many data bits then an all a string of the same length. Thus, a system that deploys a variable length matcher needs the capability to buffer large amounts of data to keep up with the variable transmission rate.

The three drawbacks of variable length matchers stated above motivate us to investigate the design of b2b matchers. For b2b matchers, the transmission rate is constant and synchronization errors and error propagation are limited by the block length. We have already seen that optimal one-to-one b2b matchers dont have good distortion vs complexity performance. Hence in this section we are interested in designing ϵ-error b2b matchers, such that $\epsilon \to 0$ asymptotically, with better distortion vs complexity performance .

Figure 4.2 and the benefits of b2b matchers discussed above motivate us loosen the constraint of one-to-one matching and to intelligently use zero error variable length matchers inside an ϵ-error b2b matcher. We show how binary ϵ-error b2b matchers can be constructed that provably achieve the same informational divergence as the zero error f2v matcher with the same complexity. The key idea is to repeatedly use the f2v matcher inside the b2b matcher. For a fixed output length, this results in underflow events and overflow events. We handle underflow events by random mapping and overflow events by casting an error. The probability of error can be made arbitrarily small by choosing the block size large enough. This corresponds to increasing the number of times the f2v matcher is applied internally and it does not effect the complexity of the b2b matcher.

In a similar manner we can construct ϵ-error b2b matchers using v2f matcher internally instead of f2v matcher[21]. Asymptotically the performance of both types of b2b matchers is the same

4.8.1 Code Construction

We use binary f2v matcher constructed in previous section repeatedly in a binary ϵ-error b2b matcher. Let the input block length for the f2v matcher be j. We can denote the f2v matcher constructed in Sec. 4.6 by $h : \{0,1\}^j \to \mathcal{X}$. We use the f2v matcher k times

in the b2b matcher hence the input length for the b2b matcher is $m = jk$. We define

$$\begin{aligned} h^k \colon \{0,1\}^m &\to \mathcal{X}^k \\ b^m &\mapsto h(b_1^j) h(b_{j+1}^{2j}) \cdots h(b_{(k-1)j+1}^{kj}). \end{aligned} \tag{4.52}$$

The total distortion for the k uses is given by

$$\frac{\mathbb{D}(P_U^k \| P_Y^{\mathcal{L}(\mathcal{X}^k)})}{\mathbb{E}[L(X^k)]} = \frac{k\,\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{k\,\mathbb{E}[\ell(X)]} \tag{4.53}$$

Define an overflow threshold

$$n \geq k\,\mathbb{E}[\ell(X)]. \tag{4.54}$$

This threshold divides the set $\mathcal{X}^k$ into two parts,

$$\begin{aligned} \mathcal{X}_{\leq} &:= \{x \in \mathcal{X}^k \colon \ell(x) \leq n\} \\ \mathcal{X}_{>} &:= \{x \in \mathcal{X}^k \colon \ell(x) > n\} \end{aligned} \tag{4.55}$$

where we write $\mathcal{X}_{\leq}$ and $\mathcal{X}_{>}$ without the super-script ^{k} for notational convenience. We define the encoder as follows.

$$\begin{aligned} f_n \colon \mathcal{X}^k &\to \{0,1\}^n \\ x &\mapsto f_n(x) = \begin{cases} x Y^{n-\ell(x)} & \text{if } x \in \mathcal{X}_{\leq} \\ x_1^n & \text{if } x \in \mathcal{X}_{>} \end{cases} \end{aligned} \tag{4.56}$$

where $Y^{n-\ell(x)}$ is a vector of $n - \ell(x)$ random variables that are iid according to P_Y and where x_1^n is x truncated to its first n entries. The mapping f_n defines a random variable $V = f_n(\mathcal{X}^k)$ that takes values in $\{0,1\}^n$. We define

$$\mathcal{V}_{\leq} := f_n(\mathcal{X}_{\leq}), \quad \mathcal{V}_{>} := f_n(\mathcal{X}_{>}). \tag{4.57}$$

The support of V can now be written as

$$\operatorname{supp} V = \mathcal{V}_{\leq} \cup \mathcal{V}_{>} \subseteq \{0,1\}^n. \tag{4.58}$$

Note that f_n is a *random* mapping. Note further that $f_n \circ h^k$ is a b2b mapping that maps

$m = jk$ bits to n bits, i.e.,

$$f_n \circ h^k \colon \{0,1\}^m \to \{0,1\}^n. \tag{4.59}$$

4.8.1.1 The Role of j and k

We discuss the intuition behind the encoder just defined. Assume n is fixed and given. Because of (4.53), the value of j controls how well the matcher output is matched to P_Y as dictated by Prop. 8. The encoder f_n is in part one-to-many (and thereby invertible) corresponding to the first case in (4.56), and in part many-to-one (which leads to errors when decoding) corresponding to the other case. For a fixed j, choosing k small decreases the many-to-one part and thereby the probability of error, but it also decreases the matching rate jk/n. Thus k parameterizes a trade-off between probability of error and matching rate.

4.8.2 ϵ-Error Matching: Analysis

4.8.2.1 Informational Divergence

Proposition 10. *The informational divergence per bit achieved by an ϵ-error b2b matcher is upper-bounded by the informational divergence per bit achieved by the internal f2v matcher , i.e.,*

$$\frac{\mathbb{D}(P_V \| P_Y^n)}{n} \leq \frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}. \tag{4.60}$$

Proof. The informational divergence can be written as

$$\mathbb{D}(P_V \| P_Y^n) = \sum_{v \in \mathcal{V}_\leq} P_V(v) \log \frac{P_V(v)}{P_Y^n(v)} + \sum_{v \in \mathcal{V}_>} P_V(v) \log \frac{P_V(v)}{P_Y^n(v)}. \tag{4.61}$$

We write the first sum as

$$\begin{aligned}
\sum_{v \in \mathcal{V}_\leq} P_V(v) \log \frac{P_V(v)}{P_Y^n(v)} &= \sum_{\substack{x \in \mathcal{X}_\leq \\ y \in \{0,1\}^{n-\ell(x)}}} 2^{-m} P_Y^{n-\ell(x)}(y) \log \frac{2^{-m} P_Y^{n-\ell(x)}(y)}{P_Y^{\ell(x)}(x) P_Y^{n-\ell(x)}(y)} \\
&= \sum_{x \in \mathcal{X}_\leq} 2^{-m} \log \frac{2^{-m}}{P_Y^{\ell(x)}(x)} \Big[\sum_{y \in \{0,1\}^{n-\ell(x)}} P_Y^{n-\ell(x)}(y) \Big] \\
&= \sum_{x \in \mathcal{X}_\leq} 2^{-m} \log \frac{2^{-m}}{P_Y^{\ell(x)}(x)}.
\end{aligned} \tag{4.62}$$

The second sum in (4.61) can be bounded as

$$\begin{aligned}
\sum_{v\in\mathcal{V}_>} P_V(v)\log\frac{P_V(v)}{P_Y^n(v)} &= \sum_{v\in\mathcal{V}_>}\Big[\sum_{x\in\mathcal{X}\colon x_1^n=v} 2^{-m}\Big]\log_2\frac{\sum_{x\in\mathcal{X}\colon x_1^n=v}2^{-m}}{P_Y^n(x_1^n)}\\
&\overset{(a)}{=} \sum_{v\in\mathcal{V}_>}\Big[\sum_{x\in\mathcal{X}\colon x_1^n=v} 2^{-m}\Big]\log\frac{\sum_{x\in\mathcal{X}\colon x_1^n=v}2^{-m}}{\sum_{x\in\mathcal{X}\colon x_1^n=v}P_Y^{\ell(x)}(x)}\\
&\overset{(b)}{\le} \sum_{v\in\mathcal{V}_>}\sum_{x\in\mathcal{X}\colon x_1^n=v} 2^{-m}\log\frac{2^{-m}}{P_Y^{\ell(x)}(x)}\\
&= \sum_{x\in\mathcal{X}_>} 2^{-m}\log\frac{2^{-m}}{P_Y^{\ell(x)}(x)} \qquad (4.63)
\end{aligned}$$

where we have equality in (a) $\mathcal{X}$, and consequently $\mathcal{X}^k$, in complete tree for the f2v matcher constructed in Sec. 4.6. The inequality in (b) follows by the log-sum inequality [10]. Using (4.62) and (4.63) in (4.61), we get

$$\begin{aligned}
\mathbb{D}(P_V\|P_Y^n) &\le \sum_{x\in\mathcal{X}_\le} 2^{-m}\log\frac{2^{-m}}{P_Y^{\ell(x)}(x)} + \sum_{x\in\mathcal{X}_>} 2^{-m}\log\frac{2^{-m}}{P_Y^{\ell(x)}(x)}\\
&= \sum_{x\in\mathcal{X}_\le\cup\mathcal{X}_>} 2^{-m}\log\frac{2^{-m}}{P_Y^{\ell(x)}(x)}\\
&= \mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{X}^k)}). \qquad (4.64)
\end{aligned}$$

Thus, we have

$$\begin{aligned}
\frac{\mathbb{D}(P_V\|P_Y^n)}{n} &\overset{(a)}{\le} \frac{\mathbb{D}(P_U^k\|P_Y^{\mathcal{L}(\mathcal{X}^k)})}{n}\\
&\overset{(b)}{\le} \frac{\mathbb{D}(P_U^k\|P_Y^{\mathcal{L}(\mathcal{X}^k)})}{\mathbb{E}[L(X^k)]}\\
&= \frac{k\,\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{X})})}{k\,\mathbb{E}[\ell(X)]}\\
&= \frac{\mathbb{D}(P_U\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} \qquad (4.65)
\end{aligned}$$

where (a) follows by (4.64) and where (b) follows by (4.54). This concludes the proof of the proposition. □

The statement of Prop. 10 can be intuitively explained by the definition (4.56) of our encoder. For input strings causing underflow, we randomly generate the missing bits

according to the target distribution P_Y. Thus, these bits do not contribute to the informational divergence. For the input strings causing overflow, we use a many-to-one mapping by truncation. This can only decrease the informational divergence because of the convexity of I-divergence.

4.8.3 Probability of Error

We use letter typicality on B^j. Assume $b^{jk} \in \mathcal{T}_\epsilon^k(B^j)$ and $x^k = f_n(b^{jk})$. By the typical average lemma [22, p. 26],

$$\frac{L(x^k)}{k} \leq (1+\epsilon)\,\mathbb{E}[\ell(X)] \tag{4.66}$$

We choose

$$n = (1+\epsilon)k\,\mathbb{E}[\ell(X)]. \tag{4.67}$$

We define the decoder as

$$\begin{aligned} \varphi_n \colon \mathcal{V}_{\leq} \cup \mathcal{V}_{>} &\to \{0,1\}^{jk} \\ v \mapsto \hat{b}^{jk} = \varphi_n(v) &= \begin{cases} (f_n \circ h^k)^{-1}(v) & \text{if } v \in \mathcal{V}_{\leq} \\ \text{error} & \text{if } v \in \mathcal{V}_{>}. \end{cases} \end{aligned} \tag{4.68}$$

By (4.66) and (4.67), an error can only occur if $B^{jk} \notin \mathcal{T}_\epsilon^k(B^j)$, i.e., if the binary sequence to be encoded is not typical. The probability of error is thus bounded by

$$\begin{aligned} \Pr(B^{jk} \neq \hat{B}^{jk}) &\leq \Pr(B^{jk} \notin \mathcal{T}_\epsilon^k(B^j)) \\ &\leq \delta_\epsilon(P_{B^j}, k). \end{aligned} \tag{4.69}$$

This probability can be made arbitrarily small by choosing k large.

4.8.4 Rate

The rate is

$$\begin{aligned}\frac{m}{n} &= \frac{kj}{(1+\epsilon)k\,\mathbb{E}[\ell(X)]} \\ &= \frac{j}{(1+\epsilon)\,\mathbb{E}[\ell(X)]} \\ &= \frac{1}{1+\epsilon}\cdot\frac{\mathbb{H}(P_U)}{\mathbb{E}[\ell(X)]} \\ &= \frac{1}{1+\epsilon}\cdot\frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]}.\end{aligned} \tag{4.70}$$

Thus, using Entropy-Divergence Theorem (Prop. 1)

$$\begin{aligned}\frac{m}{n} =& \frac{1}{1+\epsilon}\cdot\frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} \\ \overset{j\to\infty}{\to}& \frac{1}{1+\epsilon}\,\mathbb{H}(P_Y).\end{aligned} \tag{4.71}$$

The value of ϵ can be chosen arbitrarily small. This shows that our matcher can asymptotically achieve the maximum entropy rate of $\mathbb{H}(P_Y)$.
We illustrate the trade-off between informational divergence, rate, and probability of error of an ϵ-error b2b matcher by an example. We consider the target distribution P_Y with $P_Y(0) = 0.2$ and $P_(1) = 0.8$. The overflow threshold of the b2b matcher is $n = 58\,320$. In Fig. 4.3 the trade-off between rate and probability of error is displayed for internal f2v matchers with $j = 5$ (red curve) and $j = 10$ (blue curve). In horizontal direction, the gap between the rate and the target entropy $\mathbb{H}(P_Y)$ is displayed. In vertical direction, the probability of error is shown. As we can see, for $j = 10$, we need to use a lower rate to achieve the same probabilities of error as for $j = 5$. However, via Prop. 10, we can see from Fig. 4.3 that the matcher with $j = 10$ achieves a smaller informational divergence per bit than the matcher with $j = 5$.

4.9 Relation to Lossless Source Coding

In lossless source coding for DMS P_Y, the aim is to reversibly transform the input into a bit sequence such that the rate, i.e., expected number of output bits per input symbol is minimized. The lower bound on rate of a source code is $\mathbb{H}(P_Y)$ and various codes including Huffman code and Tunstall code, achieve this limit asymptotically. Furthermore we know from [23] that the output of an ideal source encoder which compresses the input sequence to the entropy limit is same as of DMS P_B, i.e., uniform i.i.d bits. Hence an ideal

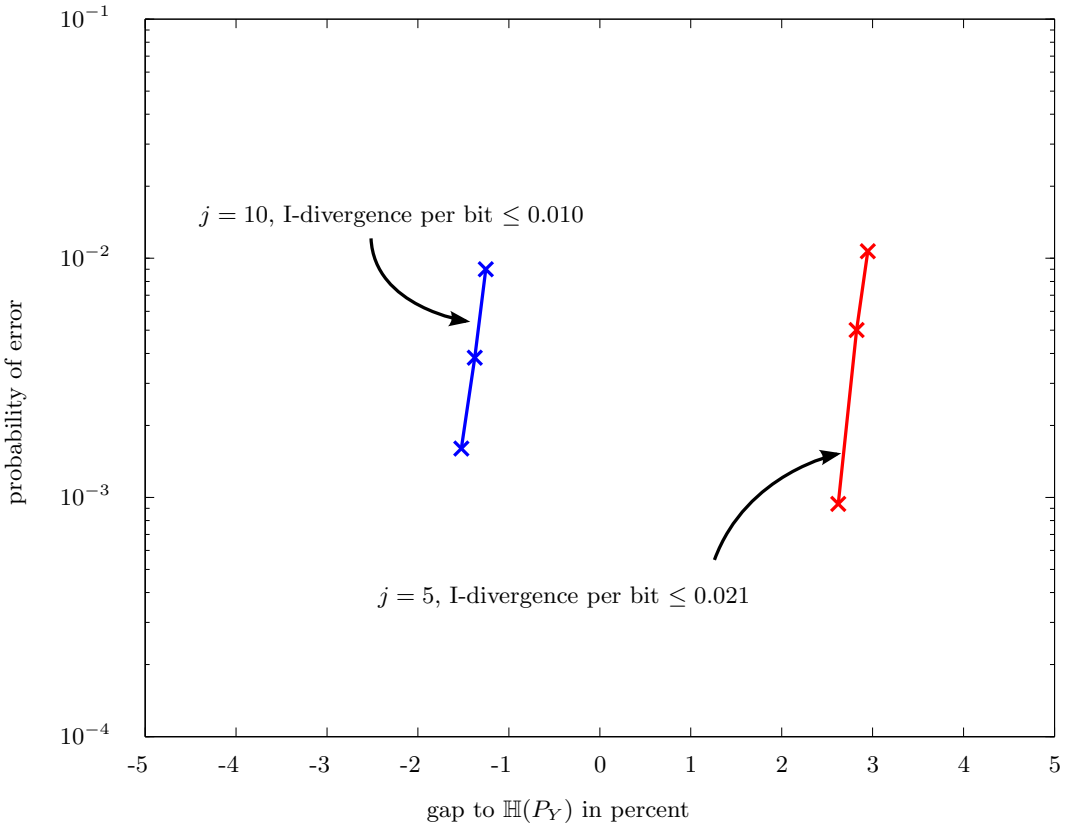

Figure 4.3: The trade-off between rate and probability of error is shown for the target distribution P_Y with $P_Y(0) = 0.2$, $P_Y(1) = 0.8$. The output block length of the ϵ-error b2b matcher is $n = 58\,320$. Internally, a f2v matcher with $j = 5$ (red curve) and $j = 10$ (blue curve) is used.

source encoder reversibly transforms a DMS P_Y to the DMS P_B. On the other hand, an ideal matcher reversibly tranforms a DMS P_B to DMS P_Y. Hence lossless source coding and distribution matching appear to be opposite procedures leading to the idea that possibly optimal source codes may lead to good distribution matching techniques. In the following discussion we look at the two famous optimal variable length lossless source coding techinques Huffman coding and Tunstall coding and see their relation to their distribution matching counterparts, i.e., GHC and Algo. 1.

4.9.1 Geometric Huffman Coding and Huffman Coding

Detailed discussion on this can be found be [6].An example is presented in [6] for comparison between Huffman coding and GHC matcher for v2f matching which shows the suboptimality of Huffman coding for v2f distribution matching.

4.9.2 Fixed-to-Variable length Distribution Matching and Tunstall Coding

Let $\mathcal{X}$ be a complete $|\mathcal{Y}|$-ary tree (in this section $|\mathcal{Y}| = 2$). Let $|\mathcal{L}(\mathcal{X})| = 2^m$. We can use this tree for v2f lossless source encoding for some DMS P_Y where we assign a unique m bit output codeword to each $x \in \mathcal{X}$. The resulting entropy rate at the output of the source encoder can be rewritten as

$$\begin{aligned}\frac{1}{m}\,\mathbb{H}(P_Y^{\mathcal{L}(\mathcal{X})}) &= \sum_{i\in\mathcal{L}(\mathcal{X})} P_Y^{\mathcal{L}(\mathcal{X})}(i)[-\log P_Y^{\mathcal{L}(\mathcal{X})}(i)] \\ &= \frac{1}{m}\sum_{i\in\mathcal{L}(\mathcal{X})} P_Y^{\mathcal{L}(\mathcal{X})}(i)[-\log P_Y^{\mathcal{L}(\mathcal{X})}(i) + \log P_U(i) - \log P_U(i)] \\ &= 1 - \frac{1}{m}\,\mathbb{D}(P_Y^{\mathcal{L}(\mathcal{X})}\|P_U) \end{aligned} \tag{4.72}$$

For v2f source coding we know that the objective is to maximize the expected input length $\mathbb{E}_{P_Y^{\mathcal{L}(\mathcal{X})}}[L(X)]$. Note that the expectation here is w.r.t to $P_Y^{\mathcal{X}}$ in contrast to (4.20) where it is w.r.t P_X. The reason is because for v2f source encoding $P_Y^{\mathcal{L}(\mathcal{X})}$ is the source distribution not the target distribution. Using the same procedure for $P_Y^{\mathcal{L}(\mathcal{X})}$ as we did in deriving (3.2) we get

$$\mathbb{H}(P_Y^{\mathcal{L}(\mathcal{X})}) = \mathbb{H}(P_Y)\,\mathbb{E}_{P_Y^{\mathcal{L}(\mathcal{X})}}[L(X)] \tag{4.73}$$

Combining (4.72) and (4.73) we conclude that the v2f lossless source coding problem can be written as

Table 4.1: Comparison of v2f source coding and f2v distribution matching: P_Y: $P_Y(0) = 0.615, P_Y(1) = 0.385$; $m = 2$

	Tunstall on P_Y	Alg. 1 on P_Y
v2f source encoder	$00 \mapsto 00$ $01 \mapsto 01$ $10 \mapsto 10$ $11 \mapsto 11$	$1 \mapsto 00$ $01 \mapsto 01$ $001 \mapsto 10$ $000 \mapsto 11$
redundancy $\frac{\mathbb{D}(P_Y^{\mathcal{L}(\mathcal{X})} \| P_U)}{m}$	0.038503	0.04176
f2v distribution matcher	$00 \mapsto 00$ $01 \mapsto 01$ $10 \mapsto 10$ $11 \mapsto 11$	$00 \mapsto 1$ $01 \mapsto 01$ $10 \mapsto 001$ $11 \mapsto 000$
I-divergence per bit $\frac{\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}$	0.039206	0.037695

$$\min_{\mathcal{X}:|\mathcal{L}(\mathcal{X})|=2^m} \frac{\mathbb{D}(P_Y^{\mathcal{L}(\mathcal{X})} \| P_U)}{m} \tag{4.74}$$

This problem is solved by Tunstall Coding on P_Y.In the optimization problems for binary Tunstall coding and binary fixed to variable length distribution matching the arguments of the informational divergence in the optimization problems are interchanged. Moreover the normalization factor is also different in the objective functions. Consider the DMS P_Y $P_Y(0) = 0.615, P_Y(1) = 0.385$. We calculate the optimal binary v2f source encoder with blocklength $m = 2$ by applying Tunstall coding to P_Y. The resulting encoder is displayed in the 1st column of Table 4.1. Using the source decoder as a distribution matcher results in an I-divergence per bit of 0.039206 bits. Next, we use Alg. 1 to calculate the optimal f2v matcher for P_Y. The resulting mapping is displayed in the 2nd column of Table 4.1. The achieved informational divergence per bit is 0.037695 bits, which is smaller than the value obtained by using the source decoder.
This example shows the suboptimality of binary Tunstall coding for f2v matching.

5 Resolution Coding for target Distributions

In this chapter we look at the second variant of the "basic problem" known as *Resolution Coding for Target Distribution*. In Sec. 5.1 we define the problem and highlight its difference to Distribution Matching. In Sec. 5.2 we prove a lower bound on the achievable rates for resolution coding. Then we discuss the related literature. We proceed by proposing optimal b2b and various variable length encoders under additional constraints and show achievability for all of these encoders. Finally we give an perforamce comparison of all these proposed encoders and discuss the relation between source coding for DMS and resolution coding for simulation of a DMS. Parts of Sec. 5.2, Sec. 5.4 and Sec. 5.5 have been taken from [24] and [9].

5.1 Resolution Coding for Target Distribution

Resolution Coding for a Target Distribution is a variant of the "basic problem" with the following additional restriction

- f is restricted to be a deterministic mapping.

In contrast to Distribution Matching we dont have any reversibility constraint.
We will use 3 different rate definitions. They are defined as follows

$$\begin{aligned} R_k &= \frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \\ R_{hv} &= \frac{\mathsf{R}(P_X)}{\mathbb{E}[\ell(X)]} \\ \bar{R} &= \frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} \end{aligned} \tag{5.1}$$

For the rest of this chapter if a statement is valid for a generic "R", it holds for each of $\{R_k, R_{hv}, \bar{R}\}$. For any probability distribution P, $\mathsf{R}(P) \geq \mathbb{H}(P)$ [3]. Hence $R_{hv} \geq \bar{R}$ for

any variable length encoder. For variable length encoders with a deterministic mapping

$$\begin{aligned}\mathbb{E}[\ell(U)] &\overset{(a)}{=} \mathbb{H}(P_U) \\ &\overset{(b)}{=} \mathbb{H}(P_{U,f(u)}) \\ &\geq \mathbb{H}(P_X)\end{aligned} \tag{5.2}$$

where a follows from (3.2) and b follows from the chain rule for entropy and the fact that $X = f(U)$. This result implies that $R \geq \bar{R}$ for any variable length encoder with deterministic mapping f. Hence the three rate definitions are related for variable length encoders with deterministic mapping f as follows

$$\{R_k, R_{hv}\} \geq \bar{R} \tag{5.3}$$

We use the following three distortion measures.

$$\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})}) \overset{(a)}{\geq} \mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}) \overset{(b)}{\geq} \frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} \tag{5.4}$$

where $\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})})$ corresponds to exact random number generation, i.e., the distortion is 0 if $P_X = P_Y^{\mathcal{L}(\mathcal{X})}$ otherwise it is ∞. We have (b) because $\mathbb{E}[\ell(X)] \geq 1$ and (a) follows from the fact that $\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})})$ is always ∞ except when $P_X = P_Y^{\mathcal{L}(\mathcal{X})}$ in which case also $\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}) = 0$ For the rest of this chapter if a statement is true for a generic distortion "D", it is valid for each of $\left\{\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})}), \mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}), \frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right\}$.

Definition 9. *A resolution rate* R *is called* achievable *if there exists a family of encoders* $(\mathcal{U}_q, \mathcal{X}_q, f_q)$ *with resolution rate* R_q *such that*

$$\begin{aligned}D_q &\overset{q\to\infty}{\to} 0 \\ R_q &\overset{q\to\infty}{\to} \mathrm{R}.\end{aligned}$$

where R_q *and* D_q *represent one of three possible rate and distortion measures. Whenever we mention an achievability result or converse, it will mentioned according to which* $\{$*rate, distortion*$\}$ *pair is it valid for. If it is valid for any of the three rate(distortion) measures then in we will mention* $R(D)$ *in the pair.*

Problem Statement:

To find the set of achievable rates for a target distribution P_Y.

5.2 Converse

In this section we prove a lower bound on the achievable rates. To make the lower bound most general we show it for the weakest conditions ,i.e. ,for $\frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}$ and $\bar{R}$. This will lead to a converse which is valid for any pair (R, D) such that $R \in \{R_k, R_{hv}, \bar{R}\}$ and $D \in \left\{\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})}), \mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})}), \frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right\}$.
From Entropy-Divergence Theorem(Prop. 1) we know

$$\begin{aligned}\frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} &\leq \epsilon \\ \Rightarrow \left|\frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} - \mathbb{H}(P_Y)\right| &\leq \delta(\epsilon)\end{aligned} \tag{5.5}$$

where $\delta(\epsilon) \xrightarrow{\epsilon\to 0} 0$. Note that $\bar{R} = \frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]}$. This leads to the following statement

Proposition 11.

$$\begin{aligned}\frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} &\leq \epsilon \\ \Rightarrow \bar{R} &\geq \mathbb{H}(P_Y) - \delta(\epsilon)\end{aligned} \tag{5.6}$$

such that $\delta(\epsilon) \xrightarrow{\epsilon\to 0} 0$. *Hence no* $R < \mathbb{H}(P_Y) - \gamma$ *is achievable* $\forall \gamma > 0$.

Remark 6. *Note that we can replace* $\frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}$ *by* $\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})$ *or* $\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})})$ *in* (5.6) *and it is still valid because of the relation mentioned in* (5.4). *Moreover we can replace* $\bar{R}$ *by* R_k *or* R_{hv} *and still the statement holds because of the relation mentioned in* (5.3).

5.3 Literature

5.3.1 Approximation of Output Statistics

In approximation of output statistics we introduce a channel after the encoder in the setup shown in Fig. 3.1 and our aim is to approximate some target process at the *output* of the channel using a deterministic encoder. We restrict our discussion to DMC and target processes which correspond to a DMS for the sake of brevity and conformity with the discussion in the rest of this thesis.
In [7], Wyner has proved the following converse for b2b deterministic encoders.

Proposition 12. *For a DMC specified by $P_{Y|W}$ and a target DMS P_Y, no $\bar{R} < \mathbb{I}(P_W; P_{Y|W}) - \gamma$ is achievable w.r.t. $\frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}$ $\quad \forall \gamma > 0$.*

Note that the converse is for the weakest pair, i.e., $\left(\bar{R}, \frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$ similar to the converse stated in Sec. 5.2 and hence is valid for any pair (R, D).
In [3], Han and Verdu have shown achievability for the pair $\left(R_{hv}, \frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$ using b2b deterministic encoders. Note that this automatically implies achievability for the pair $\left(\bar{R}, \frac{\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$. In [5], the authors have strengthened the result and have shown achievability for $\left(R, \mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})})\right)$. In both [3] and [5], the discussion is based on random coding arguments and and hence only the *existence* of b2b deterministic encoders that can achieve the lower bound has been shown. No insights into the construction of practical encoders has been provided.
In the case when $P_{Y|W}$ is the identity channel, the problem of approximating output statistics reduces to resolution coding for target distribution. Prop. 12 becomes a special case of Prop. 11 since the latter is valid for any variable length deterministic encoder whereas Wyner's converse is restricted to b2b deterministic encoders.

Remark 7. *The discussion in [3] for approximation of output statistics has been done in a much more general context where the channel can be any stochastic matrix and any target process at the output. The converse presented in [3] characterisitizes the channel and is independent of the target process and hence is weaker than the converse presented here and in [7] since these are also valid for individual target DMS. Furthermore in [3], they have also discussed approximation of output statistics when the distortion is measured in terms of variational distance.*

5.3.2 Exact Random Number Generation

Exact random number generation refers to the case when we use $\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})})$ as the distortion measure, i.e., we focus our attention only to those encoders with $P_X = P_Y^{\mathcal{L}(\mathcal{X})}$. In this case since the distortion is zero for all encoders of interest, the only important parameter is the rate of the encoder. For exact random number generation

$$\begin{aligned} \bar{R} &= \frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} \\ &= \frac{\mathbb{E}[\ell(X)]\,\mathbb{H}(P_Y)}{\mathbb{E}[\ell(X)]} \\ &= \mathbb{H}(P_Y) \end{aligned} \tag{5.7}$$

Moreover

$$R_{hv} = \frac{\mathsf{R}(P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} \tag{5.8}$$

Hence $\bar{R}$ is independent of the encoder and R_{hv} is independent of $\mathcal{U}$ and the deterministic mapping f. Hence we will focus on R_k for the rest of the discussion regarding exact random number generation to take into account the construction of the encoder. The converse stated in Sec. 5.2 is valid for exact random number generation, but we can develop a more precise converse for R_k. Continuing the chain of inequalities in (5.2), we get

$$\begin{aligned} \mathbb{E}[\ell(U)] &\geq \mathbb{H}(P_X) \\ &\overset{(a)}{=} \mathbb{H}(P_Y^{\mathcal{L}(\mathcal{X})}) \\ &\overset{(b)}{=} \mathbb{E}[\ell(X)]\, \mathbb{H}(P_Y) \end{aligned} \tag{5.9}$$

where (a) is because $P_X = P_Y^{\mathcal{L}(\mathcal{X})}$ for exact random number generation and (b) follows from the application of Leaf Entropy Theorem and the Path Length Lemma [12]. Hence we have

$$R_k \geq \mathbb{H}(P_Y) \tag{5.10}$$

Note that we cannot use similar method to prove a converse for *approximate* random number generation since (a) in (5.9) is not true in general. For exact random number generation (5.9) we can only approach the entropy limit on achievable rates from above in contrast to approximate random number generation where we can in principle approach it both from above and below as Prop. 11 suggests.

In [1], Knuth and Yao have presented the encoder for exact random number generation that minimizes R_k for fixed output block length n for any target DMS P_Y. It is a v2f encoder. We will call it the Knuth encoder for the rest of the chapter .Denote the target distribution by P_Y. We can consider that the output corresponding to $P_Y(i)$ is i without loss of generality for $1 \leq i \leq |\operatorname{supp}(P_Y)|$. The following algorithm defines the Knuth encoder

Algorithm 2.

$i = 1$

repeat while $i \leq |\operatorname{supp}(P_Y)|$

1. $j = 1$

2. $\mathcal{S}_i = \emptyset$
3. repeat while $j < \infty$

a. If $\lfloor 2^j \cdot P(i) \rfloor - 2 \cdot \lfloor 2^{j-1} \cdot P(i) \rfloor = 1$ then choose a binary string u s.t. $u \notin \bigcup_{k=1}^{i} \mathcal{S}_k$ and $\ell(u) = j$. Add this u to $\mathcal{S}_i$.

b. $j \leftarrow j + 1$

c. Assign all strings in set S_i to output i.

d. 4. $i \leftarrow i + 1$

5. end

$\lfloor x \rfloor$ denotes the floor function. Basically Knuth algorithm uses the binary expansion of elements of P[10]. If the j bit of $P(i)$ after radix point is 1 then we map a unique binary string of length j to symbol i. Kraft inequality guarantees that a Knuth encoder can be constructed for any discrete probability distribution.We present a simple example to make the procedure more clear.

Example 3. Let $P_Y = \begin{bmatrix} \frac{5}{8} & \frac{3}{8} \end{bmatrix}$. Consider binary representation of $P_Y = \begin{bmatrix} 0.101 & 0.011 \end{bmatrix}$ and $n = 1$. Then a possible choice for u_i is

$$u_1 = \{1, 001\} u_2 = \{01, 000\}$$

For Knuth encoder it is shown in [1] that

$$\mathbb{H}(P_Y) \leq R_k \leq \mathbb{H}(P_Y) + \frac{2}{n} \tag{5.11}$$

Hence the Knuth encoder can be used to show achievability for $\{R_k, D\}$.

Remark 8. *The algorithm mentioned to construct the Knuth encoder can be applied to construct R_k optimal encoder for any distribution P, not only product distributions of the form P_Y^n.*

When using $\infty \cdot \mathbb{1}(P_X \neq P_Y^{\mathcal{L}(\mathcal{X})})$ as the distortion measure, i.e., for exact random number generation, we don't have the liberty to design encoders where we can have a tradeoff between distortion and rate. Moreover under certain additional restrictions on the deterministic encoder such as maximum delay constraint, which are not satisified by Knuth encoder and similar exact random number generation algorithms such as the interval algorithm[2], we cannot do exact random number generation and hence we have to resort to approximate random number generation. For binary f2v matchers as defined in Sec. 4.6 we have $m = \mathbb{E}[\ell(U)] = \mathbb{H}(P_U) = \mathbb{H}(P_X) = \mathsf{R}(P_X)$ because of the fixed input length and the one-to-one deterministic mapping f. This means $R_k = R_{hv} = \bar{R}$. We have also

shown in Sec. 4.6 that $R_k \to \mathbb{H}(P_Y)$ and $\frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]} \to 0$ asymptotically. Hence using this matcher we can show achievability for $\left(R, \frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$. Similarly, using GHC we can show achievability for $\left(R_k, \frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$ and $\left(\bar{R}, \frac{\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})}{\mathbb{E}[\ell(X)]}\right)$. Furthermore the mapping f for these matchers is one to one. Hence the encoder operation is invertible. This is not a requirement for resolution codes in contrast to distribution matching. This motivates us to use more restrictive distortion measure and see if we can still achieve the rate limit $\mathbb{H}(P_Y)$. In the next sections we develop various encoders for the distortion measure $\mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})})$ since it corresponds to a stronger approximation than the normalized divergence, but still gives us the freedom to trade between rate and distortion. We will show that all developed encoders can be used to prove achievability for $(R, \mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})}))$.

5.4 Block-to-Block Resolution Encoder

5.4.1 Optimal Block-to-Block Encoder

Let us characterize the set of approximating distributions that can be generated using a b2b encoder. The only freedom we have in the design of a b2b encoder of fixed input m and output blocklength n is the deterministic mapping $f : \{0,1\}^m \to \mathcal{Y}^n$. For $\mathcal{U} = \{0,1\}^m$

$$P_U(u) = 2^{-m}. \tag{5.12}$$

For any $x \in \mathcal{X}$ let $k_x = |u_x|$, where $u_x = \{u \in \{0,1\}^m : x = f(u)\}$ then

$$\begin{aligned} P_X(x) &= \sum_{u\in u_x} P_U(u) \\ &= \sum_{u\in u_x} 2^{-m} \\ &= \frac{k_x}{2^m} \end{aligned}$$

This shows that a b2b encoder can only generate 2^m-type approximating distributions. Now we show that for any 2^m-type distribution P, we can construct a b2b encoder such that $P_X = P$. We use the following procedure to construct the b2b encoder.

Algorithm 3.

---

i =1

repeat while $i \leq |\operatorname{supp}(P)|$

1. $k_i = P(i) \cdot 2^m$
2. choose k_i binary strings of length m which are not part of some $u_k \quad 1 \leq k < i$. label this set u_i
3. Define $i = f(u) \qquad \forall u \in u_i$.
4. i++ end

The total number of m bit strings are 2^m and since P is a distribution $\sum_i k_i = 2^m$ where k_i are all non-negative intergers since P is 2^m-type, hence we can find such a partition of the m bit strings. Moreover $\sum_i |u_i| = 2^m$ i.e. $\bigcup_i u_i = \{0,1\}^m$. This discussion leads to the following result.

Proposition 13. *A probability distribution P can be generated by a b2b encoder of input length m and deterministic mapping f iff P is 2^m-type.*

For any b2b encoder with input length m we have $\mathsf{R}(P_X) \leq m$. Note that we can have strict inequality here in the case when the approximating distribution is also 2^k-type for $k < m$. Hence for b2b encoders we have

$$\frac{m}{n} = R_k \geq R_{hv} \geq \bar{R}. \tag{5.13}$$

We apply Alg. 2.7.2.2 to find the informational divergence optimal 2^m-type approximation of P_Y^n. Then we use Alg. 3 to construct the encoder for this 2^m-type approximation. From the construction procedure it is clear that this b2b encoder is optimal in the sense that for a given P_Y, it minimizes informational divergence for a fixed m and n.

5.4.2 Achievability

Define a distribution over the letter typical set $\mathcal{T}_\epsilon^n(P_Y)$

$$Q(y^n) := \frac{P_Y^n(y^n)}{\Pr\{\mathcal{T}_\epsilon^n(P_Y)\}}, \quad \forall y^n \in \mathcal{T}_\epsilon^n(P_Y). \tag{5.14}$$

Denote by P the informational divergence optimal 2^m-type approximation of Q. We have

$$\begin{aligned}
\mathbb{D}(P\|P_Y^n) &= \sum_{y^n \in \operatorname{supp} P} P(y^n) \log_2 \frac{P(y^n)}{P_Y^n(y^n)} && (5.15)\\
&= \sum_{y^n \in \operatorname{supp} P} P(y^n) \log_2 \frac{P(y^n)\Pr\{\mathcal{T}_\epsilon^n(P_Y)\}}{P_Y^n(y^n)\Pr\{\mathcal{T}_\epsilon^n(P_Y)\}} && (5.16)\\
&= \mathbb{D}(P\|Q) + \log_2 \frac{1}{\Pr\{\mathcal{T}_\epsilon^n(P_Y)\}}. && (5.17)
\end{aligned}$$

The second summand goes to zero for $n \to \infty$. Using [9, Prop. 2] and [25, Theorem. 3.14] we bound

$$\mathbb{D}(P\|Q) \leq \frac{2^{n[H(P_Y)+\epsilon]}}{22^{-n[H(P_Y^n)+\epsilon]}2^{2m}} = \frac{2^{-2[m-nH(P_Y)-n\epsilon]}}{2} \tag{5.18}$$

which goes to zero for $\frac{m}{n} > H(P_Y) + \epsilon$ for some $\epsilon > 0$. Denote by P_X the 2^m-type informational divergence optimal approximation of P_Y^n obtained by using Alg. 2.7.2.2 on P_Y^n for the b2b encoder construction. By definition of P_X we have

$$\mathbb{D}(P\|P_Y^n) \geq \mathbb{D}(P_X\|P_Y^n) \tag{5.19}$$

We conclude that $\mathbb{D}(P_X\|P_Y^n) \to 0$ for $n \to \infty$ if $\frac{m}{n} > H(P_Y) + \epsilon$ for some $\epsilon > 0$. Hence this shows that b2b encoders achieve for the pair $(R, \mathbb{D}(P_X\|P_Y^{\mathcal{L}(\mathcal{X})}))$ the $\mathbb{H}(P_Y)$ lower bound on rate .

5.4.3 Performance

In Fig. 5.1 we can see the performance of the optimal b2b encoder for the target distribution $P_Y(0) = 1 - P_Y(1) = 0.211$. Each curve with a different colour corresponds to a different input block size m and points on each curve are obtained by changing the output block length n for a fixed length input m. We have plotted the distortion performance w.r.t $\frac{m}{n} = R_k$. We can see that with increasing input block size, and hence complexity, the performance of the optimal b2b encoder becomes closer to the vertical dotted line in the figure representing the $\mathbb{H}(P_Y)$ lower bound. But even for considerable sizes of the input dictionary such as 2^{12} corresponding to the blue curve there is still a significant gap between the ideal performance of a resolution encoder and the optimal b2b encoder. This shows that although b2b encoders asymptotically achieve the entropy lower bound, but in the finite length regime the performance of these encoders is not very good. This motivates us to look at variable length deterministic encoders for resolution coding.

5.5 Fixed-to-Variable Length Resolution Coding

For f2v encoder $\mathcal{U} = \{0,1\}^m$. To construct $\mathcal{X}$ with $|\mathcal{X}| = 2^p$, we apply Tunstall Coding (Sec. 2.7.1) to P_Y. Note that p does not need to be an integer but $2^p - 1$ must be divisible by $|\mathcal{Y}| - 1$. This is a weaker restriction than p being an integer hence allowing us more freedom in choosing $|\mathcal{L}(\mathcal{X})|$. We use Tunstall coding to construct $\mathcal{X}$. Tunstall coding constructs a complete tree $\mathcal{X}$ hence $P_Y^{\mathcal{L}(\mathcal{X})}$ forms a distribution over $|\mathcal{L}(\mathcal{X})|$. Furthermore

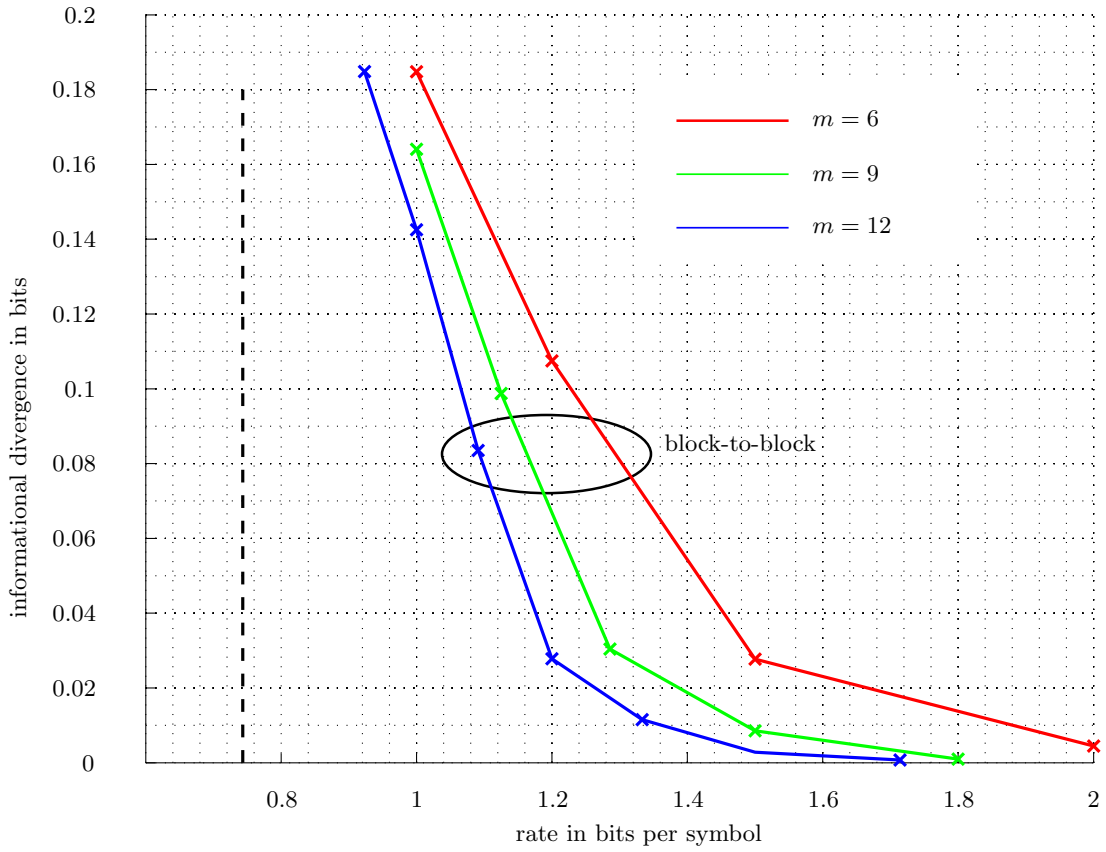

Figure 5.1: Performance of b2b resolution encoders for $P_Y(0) = 1 - P_Y(1) = 0.211$.

as shown in Sec. 2.7.1 $P_Y^{\mathcal{L}(\mathcal{X})}$ has the following properties

$$P_Y^{\mathcal{L}(\mathcal{X})}(x) \geq \mu_Y 2^{-p} \tag{5.20}$$

$$P_Y^{\mathcal{L}(\mathcal{X})}(x) \leq \frac{2^{-p}}{\mu_Y} \tag{5.21}$$

where

$$\mu_Y := \min_{a \in \operatorname{supp} P_Y} P_Y(a). \tag{5.22}$$

Note that μ_Y does only depend on P_Y and is independent of m and p.
Since we have $\mathcal{U} = \{0,1\}^m$ and a deterministic mapping $f : \mathcal{U} \to \mathcal{X}$ hence arguing the same way as for Prop. 13 we conclude that we can only generate 2^m-type approximating distributions P_X. To define the last component f of our f2v resolution encoder we do the following procedure. First we quantize $P_Y^{\mathcal{L}(\mathcal{X})}$ using Alg. 2.7.2.1 to obtain the 2^m-type approximating distribution P_X, then we apply Alg. 3 to define the mapping from $\mathcal{L}(\mathcal{U})$ to $\mathcal{L}(\mathcal{X})$ to generate P_X over $\mathcal{L}(\mathcal{X})$. The reason to use Alg. 2.7.2.1 to quantize $P_Y^{\mathcal{X}}$ is for P_X to have the property mentioned in (2.34), namely

$$P_X(x) \leq P_Y^{\mathcal{L}(\mathcal{X})}(x) + 2^{-m}, \qquad \forall x \in \mathcal{L}(\mathcal{X}). \tag{5.23}$$

This property will be useful in both bounding the rate and informational divergence for the encoder. The codebook is of size 2^p and the input of the encoder is of length m bits. We define $q := m - p$.

5.5.1 Discussion of Code Construction

- In Sec. 4.6 we saw that for binary fixed-to-variable distribution matcher, using Tunstall coding to construct $\mathcal{X}$ minimizes the un-normalized informational divergence. This motivates us to use Tunstall coding to construct $\mathcal{X}$ here as well. Moreover, the Tunstall code guarantees that for each codeword $x \in \mathcal{L}(\mathcal{X})$, the target probability $P_Y^{\mathcal{L}(\mathcal{X})}(x)$ deviates from the uniform probability $1/2^p$ at most by a factor of μ_Y, which is independent of p and m. Hence it leads to an almost uniform target distribution $P_Y^{\mathcal{L}(\mathcal{X})}$.

- Because the input is of fixed length m, it is uniformly distributed over the dictionary $\mathcal{U} = \{0,1\}^m$. A many-to-one mapping from $\mathcal{L}(\mathcal{U})$ to $\mathcal{L}(\mathcal{X})$ resolves the remaining difference between the uniform distribution over $\mathcal{L}(\mathcal{U})$ and the almost uniform target distribution over $\mathcal{L}(\mathcal{X})$.

- A many to one mapping suggests that we can improve on $\mathbb{E}[\ell(U)]$ by using a variable length input parser $\mathcal{U}$ instead of a fixed length m bit input parser. We will show in the next section that using this fixed length input parser is asmptotically optimal. This implies that the fractional gain of using a variable length input parser diminishes with increasing p and q.

Remark 9. *A similar method has been used for b2b encoder design in [3, Sec. 3B]. To get an almost uniform target distribution to approximate they have focussed on generating sequences that are typical w.r.t. P_Y. Similarly, to resolve the remaining differences among the probabilities of typical sequences they have used a many-to-one mapping from a uniform random variable formed over the fixed length input to the set of typical sequences.*

5.5.2 Performance

We now discuss the rate-distortion performance of the f2v encoder constructed in this section. First we derive bounds on informational divergence and rate in terms of μ_Y,P_X,m and p. Later we will present a numerical example to compare its performance to optimal b2b encoder in finite length regime.

- **Distortion:** We bound

$$\begin{aligned}
\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}) &= \sum_{x \in \mathrm{supp}(P_X)} P_X(x) \log \frac{P_X(x)}{P_Y^{\mathcal{L}(\mathcal{X})}(x)} \\
&\overset{(a)}{\leq} \sum_{x \in \mathrm{supp}(P_X)} P_X(x) \log \frac{P_Y^{\mathcal{L}(\mathcal{X})}(x) + 2^{-m}}{P_Y^{\mathcal{L}(\mathcal{X})}(x)} \\
&\overset{(b)}{\leq} \sum_{x \in \mathrm{supp}(P_X)} P_X(x) \log \left(1 + \frac{2^p}{2^m \mu_Y}\right) \\
&\overset{(c)}{\leq} \sum_{x \in \mathrm{supp}(P_X)} P_X(x) \frac{2^p}{2^m \mu_Y} \log e \\
&= \frac{2^p}{2^m \mu_Y} \log e \\
&= \frac{2^{-q}}{\mu_Y} \log e
\end{aligned} \tag{5.24}$$

where (a) follows from (5.23), where (b) follows from (5.20), and where we used the bound $\log(1+x) \leq x \log e$ in (c). As $2^q \to \infty$, the upper bound on the fraction $\frac{P_X(a)}{P_Y^{\mathcal{X}}(a)}$ approaches 1 $\forall a$, which shows that with increasing q, the fractional difference

between elements of the approximating distribution P_X and the target distribution $P_Y^{\mathcal{X}}$ approaches 0. Moreover, we see that for $2^q \to \infty$, we have $\mathbb{D}(P_X \| P_Y^{\mathcal{L}(\mathcal{X})}) \to 0$

- **Rate:** We cannot define a relationship between R and R_{hv} for variable length encoders in general. But for f2v encoders, using the same analysis as in Sec. 5.4.1, we conclude that $R_k \geq R_{hv}$. Combining these results we have $R_k \geq R_{hv} \geq \bar{R}$. In the rate analysis of the f2v encoder constructed in this section we therefore focus on R_k since it upper bounds the other two rates.

  We start by looking at the entropy of the approximating distribution P_X. For each $x \in \mathcal{X}$, the probability $P_X(x)$ is upper bounded by

$$\begin{aligned} P_X(x) &\overset{(a)}{\leq} P_Y^{\mathcal{L}(\mathcal{X})}(x) + 2^{-m} \\ &\overset{(b)}{\leq} \frac{2^{-p}}{\mu_Y} + 2^{-m} \end{aligned} \tag{5.25}$$

  where (a) and (b) follow from (5.23) and (5.21), respectively. We can now bound the entropy of X as follows.

$$\begin{aligned} \mathbb{H}(P_X) &= \mathbb{E}\left[\log \frac{1}{P_X(X)}\right] \\ &\geq \log \frac{1}{\max\limits_{x \in \operatorname{supp} P_X} P_X(x)} \\ &\overset{(a)}{\geq} \log \frac{1}{\frac{2^{-p}}{\mu_Y} + 2^{-m}} \\ &= p - \log\Big(\frac{1}{\mu_Y} + \frac{2^p}{2^m}\Big) \\ &= p - \log\Big(\frac{1}{\mu_Y} + 2^{-q}\Big) \end{aligned} \tag{5.26}$$

  where we used (5.25) in (a). This can also be rewritten as a bound on p, namely

$$p \leq \mathbb{H}(P_X) + \log_2\left(\frac{1}{\mu_Y} + 2^{-q}\right). \tag{5.27}$$

  Remark 10. *We know $|\mathcal{L}(\mathcal{X})| = 2^p$ and hence for the approximating distribution P_X over $\mathcal{L}(\mathcal{X})$, $\mathbb{H}(P_X) \leq p$ where the upper bound corresponds to a uniform distribution over $\mathcal{L}(\mathcal{X})$. (5.26) provides a lower bound on $\mathbb{H}(P_X)$ generated using this f2v encoder. We can see that as $2^q \to \infty$ the difference between $\mathbb{H}(P_X)$ and p is a term independent of $\mathcal{X}, m$ and p.Hence this indicates that the approximating distribution P_X asmptotically becomes closer and closer to the uniform distribution in terms of*

entropy.

Using the above results we bound R_k as follows.

$$\begin{aligned} R_k &= \frac{\mathbb{E}[\ell(U)]}{\mathbb{E}[\ell(X)]} \\ &= \frac{m}{\mathbb{E}[\ell(X)]} \\ &= \frac{p}{\mathbb{E}[\ell(X)]} + \frac{q}{\mathbb{E}[\ell(X)]} \\ &\overset{(a)}{\leq} \frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} + \frac{q + \log_2(\frac{1}{\mu_Y} + 2^{-q})}{\mathbb{E}[\ell(X)]} \end{aligned} \tag{5.28}$$

where we used (5.27) in (a).

We separately bound the two terms in (5.28). For the second term, we get

$$\begin{aligned} \frac{q + \log(\frac{1}{\mu_Y} + 2^{-q})}{\mathbb{E}[\ell(X)]} &\overset{(a)}{\leq} \frac{q + \log_2(\frac{1}{\mu_Y} + 2^{-q})}{\mathbb{H}(P_X)\frac{1}{\log|\mathcal{Y}|}} \\ &\overset{(b)}{\leq} \frac{q + \log(\frac{1}{\mu_Y} + 2^{-q})}{p - \log(\frac{1}{\mu_Y} + 2^{-q})} \log|\mathcal{Y}| \\ &\overset{\frac{q}{p}\to 0, q\to\infty}{\to} 0 \end{aligned} \tag{5.29}$$

where (a) follows from the Source Coding Theorem [23, Theo 4.1]. To understand this consider a DMS P_X. Now if we construct any D-ary source code ($D = |\mathcal{Y}|$) for this source, according to the Source Coding Theorem, the expected length of that source code per symbol will be greater than or equal to the entropy of P_X evaluated in a base $\log_D$ and $\mathbb{H}(P_X)\frac{1}{\log D}$ is exactly that. We used (5.26) in (b). For the first term in (5.28), we have by (5.24) and Entropy-Divergence Theorem(Prop. 1),

$$\frac{\mathbb{H}(P_X)}{\mathbb{E}[\ell(X)]} \overset{q\to\infty}{\to} \mathbb{H}(P_Y). \tag{5.30}$$

Using (5.29) and (5.30) in (5.28), we get

$$\lim_{\substack{\frac{q}{p}\to 0 \\ q\to\infty}} R \leq \mathbb{H}(P_Y) + \gamma. \tag{5.31}$$

for any $\gamma > 0$.

Hence as $2^q \to \infty$ and $\frac{q}{p} \to 0$ this encoder achieves any rate R_k arbitrarily close to $\mathbb{H}(P_Y)$.

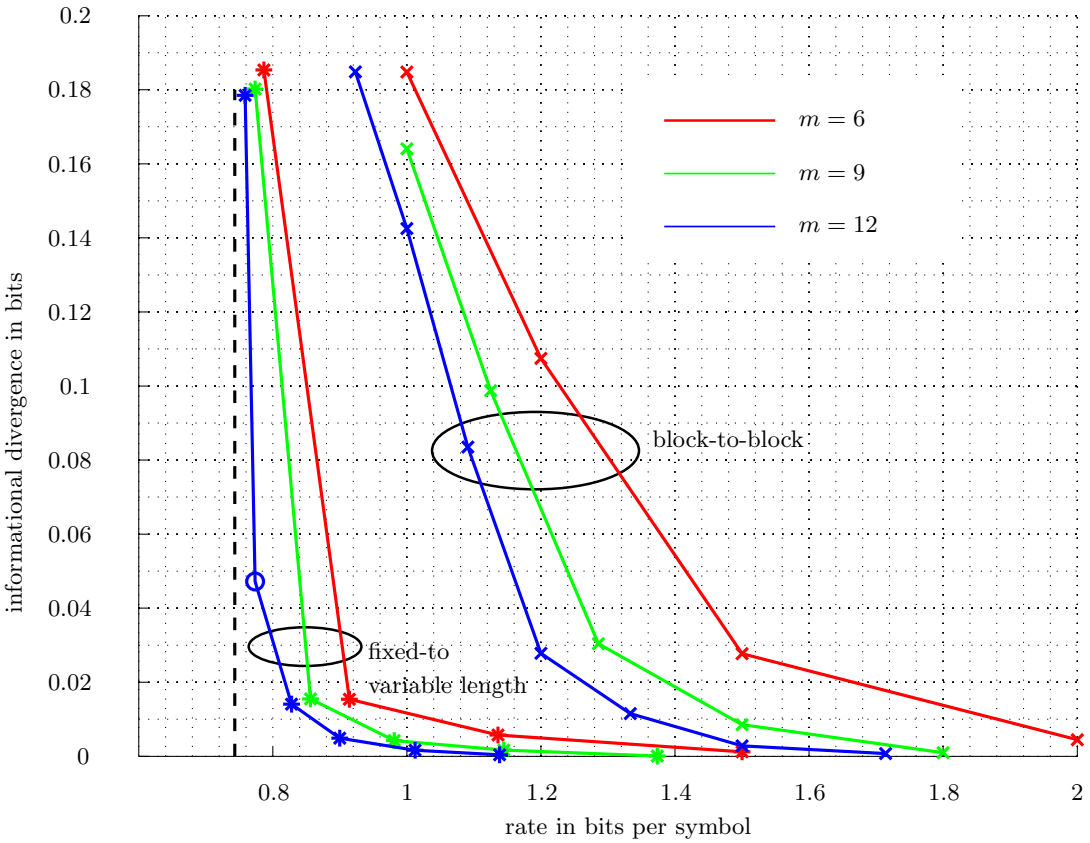

Figure 5.2: Rate vs Distrotion performance of optimal b2b and f2v resolution encoders for $P_Y(0) = 1 - P_Y(1) = 0.211$

To conclude the discussion on f2v resolution encoder we present a comparison of the rate vs distortion performance of this encoder with the optimal b2b encoder for a binarty target distribution $P_Y(0) = 1 - P_Y(1) = 0.211$ in Fig. 5.2. The simulation is done for the same parameters as in Sec. 5.4.3. We can see a significant performance improvement for the proposed f2v encoder. Moreover the circled point on the blue curve for f2v encoder corresponds to an output codebook size such that p is not an integer. This shows that we have more flexibility in terms of closely following the rate-distortion curve for this fixed-to-variable length encoder as compared to b2b encoder where you can only design encoders which operate on points corresponding to cross marked on the curves for b2b encoders.

5.6 Variable-to-fixed length resolution codes

The Knuth encoder discussed in 5.3.2 is a v2f encoder for $\mathcal{X} = \mathcal{Y}^n$. It was shown in Sec. 5.3.2 that knuth encoder can be used to show achievability for (R_k, D). Let $P_Y = \begin{bmatrix} \frac{1}{\pi} & 1 - \frac{1}{\pi} \end{bmatrix}$. Since $P_X = P_Y^n$ for Knuth encoder, $\mathsf{R}(P_X) = \infty \quad \forall n$ because of the irrational elements in P_Y^n and $R_{hv} = \infty$. This is true for any target distribution P_Y with irrational elements. Hence Knuth encoder cannot be used to show achievability for (R_{hv}, D)

Furthermore, from the design of Knuth encoder we know that depending on the target distribution P_Y, $\mathcal{U}$ may contain words of infinite length. A simple example of such a case is when we construct a knuth encoder for $P_Y = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \end{bmatrix}$ and $n = 1$. Having $u \in \mathcal{U}$ such that $\ell(u) = \infty$ i.e. binary input parsing strings are not practical to implement unless they have some special repeatitive structure. Moreover these strings also can lead to infinitely long delay in terms of the number of input bits required to generate a block of symbols at the output, although with vanishing probability.

The two problems just mentioned for Knuth encoder motivate us to look at a v2f encoder construction that tackles problems.

5.6.1 Maximum Delay Encoder

We construct a v2f encoder where we restrict the maximum length of any $u \in \mathcal{U}$ i.e. $\ell(u) \leq m$. Having a maximum length constraint on the elements of $\mathcal{U}$ allows us to control the implementation complexity (in terms of the strings in $\mathcal{U}$ that need to be stored) as well as the maximum delay (in terms of the number of inputs before the encoder outputs a block of symbols). To show that having this additional constraint also allows us to have a control over the resolution of the approximating distribution P_X, and hence R_{hv} of the encoder, we will now characterize the set of distributions that can be generated by such encoders. This characterization will also help us in defining the construction rule for such encoders. We know that

$$P_U(u) = 2^{-\ell(u)} \qquad \forall u \in \mathcal{L}(\mathcal{U}) \tag{5.32}$$

For any $x \in \mathcal{L}(\mathcal{X})$

$$\begin{aligned} P_X(x) &= \sum_{u \in u_x} P_U(u) \\ &= \sum_{u \in u_x} 2^{-\ell(u)} \\ &= \sum_{u \in u_x} \frac{2^{m-\ell(u)}}{2^m} \\ &= \frac{1}{2^m} \sum_{u \in u_x} 2^{m-\ell(u)} \\ &= \frac{M_x}{2^m} \end{aligned} \tag{5.33}$$

where $u_x = \{u \in \mathcal{U} : x = f(u)\}$ and M_x is a non negative integer between 0 and 2^m. $M_x \geq 0$ since $\ell(u) \leq m \quad \forall u \in u_x$. Moreover $M_x \leq 2^m$ since P_X is a distribution. This shows that for any v2f encoder P_X is 2^m-type similar to the case of b2b encoders. Hence

$$\mathsf{R}(P_X) \leq m \tag{5.34}$$

So the maximum length m upper bound the resolution of the approximating distribution P_X. Now we show that for any 2^m-type distribution P we can construct a variable to fixed length encoder with maximum length constraint such that $P_X = P$. For this purpose we apply Algo. 3 to P to construct a b2b encoder having input block length m with $P_X = P$. Since b2b encoders with input length m are a subset of v2f encoders satisfying maximum length constraint m hence this completes the proof of the following proposition

Proposition 14. *A probability distribution P can be generated by a v2f encoder with maximum delay constraint m and deterministic mapping f iff P is 2^m-type.*

After having characterized the set of distribution that can be generated using a mximum delay encoder we design an encoder to approximate any target distribution P_Y in a two step process similar to the one used to construct optimal b2b encoder in Sec. 5.4. First we find the 2^m-type informational divergence optimal quantization P for P_Y^n using Alg.2.7.2.2. We then apply the following algorithm to P

Algorithm 4.

---

i =1

repeat while $i \leq |\operatorname{supp}(P)|$

1. $k_i = P(i)$
2. $u_i = \emptyset$

3. $j = 1$
4. **Repeat while** $j \leq m$

a. If $\lfloor 2^j \cdot P(i) \rfloor - 2 \cdot \lfloor 2^{j-1} \cdot P(i) \rfloor = 1$ then choose a binary string u s.t. $u \notin \bigcup_{k=1}^{i} u_k$ and $\ell(u) = j$. Add this u to u_i.

b. j++

5. Define $i = f(u) \qquad \forall u \in u_i$
6. i++

Since P is a 2^m-type distribution, hence any element of it can be completely represented by its m bits after the binary point. Moreover $\mathcal{U} = \bigcup_i u_i$ forms a complete binary dictionary since P is a distribution and all its elements are completely represented by m bits after the binary point. The deterministic mapping f is defined by the partitions u_i. For each i all the elements of u_i are mapped to $i \in \mathcal{Y}^n$. This completes the construction of v2f encoder with maximum delay constraint.
For this encoder we have

$$R_k \leq \frac{m}{n} \tag{5.35}$$

$$R_{hv} \leq \frac{m}{n} \tag{5.36}$$

where the first equation follows from the fact that for any $u \in \mathcal{L}(\mathcal{U})$, $\ell(u) \leq m$ and second equation follows from (5.34). Moreover From Sec. 5.4 we know that for $\frac{m}{n} > \mathbb{H}(P_Y) + \epsilon$, for any $\epsilon > 0$ and $n \to \infty$, we have $\mathbb{D}(P \| P_Y^n) \to 0$ where P represents the 2^m-type informational divergence optimal approximation of P_Y^n. Note that for this v2f encoder with maximum length constraint the approximating distribution P_X is equal to P, i.e., the 2^m informational divergence optimal approximation of P_Y^n. Hence based on this discussion we conclude that this encoder can be used to show achievability for $(R, \mathbb{D}(P_X \| P_Y^n))$. Note that in constrast to knuth encoder we now also have the achievability result for R_{hv} but we have loosened the restriction of exact random number generation to approximate random number generation with un-normalized informational divergence distortion.

Remark 11. *As evident from the design procedure, for a P_Y^n which is 2^m-type this encoder and knuth encoder will be identical.*

Remark 12. *This encoder has minimum R_k among all the v2f encoders under maximum delay constraint which generate the approximating distribution P minimizing the informational divergence among the set of all possible 2^m-type distributions.*

Remark 13. *The only difference between this v2f encoder with maximum length constraint and optimal b2b encoder presented in Sec.5.4 is that we have used Algo.5 instead*

of Algo.3 to construct $\mathcal{U}$. This leads to a better R_k but the distortion for both the encoders is the same since both have same P_X. Moreover same P_X implies that both encoders have same R_{hv} and $\bar{R}$.

5.6.2 Finite State Generator Encoder

To have a $\mathcal{U}$ such that $\ell(u) \leq m \quad \forall u \in \mathcal{L}(\mathcal{U})$ is very restrictive in terms of the set of approximating distributions that can be generated. Using such encoders, as we saw in the previous section, we can only generate 2^m-type distributions which is the same as for b2b encoders. We want to ease this restriction, allowing $\mathcal{U}$ such that $\exists u \in \mathcal{L}(\mathcal{U})$ for which $\ell(u) = \infty$, in such a way that $\mathcal{U}$ has some special structure making it suitable for implementation. In this regard Finite State Generators(FSG) discussed in [1] are a suitable candidate. In this section we will focus on a special case of Finite State Generators which is specially suitable for implementation and has same implementation complexity as a v2f encoder with maximum length constraint although this encoder will have $u \in \mathcal{U}$ such that $\ell(u) = \infty$.

The architecture of an FSG encoder is as follows. Since the output is fixed length we have $\mathcal{X} = \mathcal{Y}^n$. Let $\mathcal{U}_1$ be some complete binary dictionary such that $\forall u \in \mathcal{L}(\mathcal{U}_1) \quad \ell(u) \leq m$. Let $\mathcal{L}'(\mathcal{U}_1)$ and $\mathcal{L}''(\mathcal{U}_1)$ form a partition of $\mathcal{L}(\mathcal{U}_1)$. $\mathcal{U}$ is constructed such that

$$\begin{aligned}\mathcal{L}(\mathcal{U}) = \Big\{ & u : u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix} \\ & \text{such that } 1 \leq N \leq \infty, u_N \in \mathcal{L}'(\mathcal{U}_1) \\ & \text{and } u_i \in \mathcal{L}''(\mathcal{U}_1) \forall 1 \leq i \leq n-1 \Big\}\end{aligned} \tag{5.37}$$

where $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}$ represents the concatenation of the binary strings u_1 to u_n. For example if $u_1 = 01$ and $u_2 = 1$ then $\begin{bmatrix} u_1 & u_2 \end{bmatrix} = 011$. Note that specifying $\mathcal{L}(\mathcal{U})$ completely chahracterizes $\mathcal{U}$. The main concept behind the construction of $\mathcal{U}$ is that we start with $\mathcal{U}_1$ and root the same $\mathcal{U}_1$ tree at each of the nodes in $\mathcal{L}''(\mathcal{U}_1)$ and in every of these subtrees rooted at one of the nodes in $\mathcal{L}''(\mathcal{U}_1)$ we again repeat the same process and continue this infinitely. This leads to $\mathcal{U}$. Note that for $\mathcal{U} = \mathcal{U}_1$ we have the v2f encoder with maximum delay constraint. Hence v2f encoder with maximum delay constraint is a special case of this type of FSGs.

The mapping f has the following restriction. if $u', u'' \in \mathcal{L}(\mathcal{U})$ such that

$$u' = \begin{bmatrix} u'_1 & u'_2 & \cdots & u'_{N_1} \end{bmatrix}$$
$$u'' = \begin{bmatrix} u''_1 & u''_2 & \cdots & u''_{N_2} \end{bmatrix}$$

where $u'_1, \cdots, u'_{N_1-1}, u''_1, \cdots, u''_{N_2-1} \in \mathcal{L}''(\mathcal{U}_1)$ and $u'_{N_1} = u''_{N_2} \in \mathcal{L}'(\mathcal{U}_1)$ then $f(u') = f(u'')$. Basically for any $u = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix} \in \mathcal{L}(\mathcal{U})$, the value of the deterministic mapping $f(u)$ only depends upon the last component of the concatenation $u_N \in \mathcal{L}'(\mathcal{U}_1)$ and is independent of the rest of the components from $\mathcal{L}''(\mathcal{U}_1)$. Such a construction of $\mathcal{U}$ and the deterministic mapping leads to the following recursive expression for the elements of P_X

$$P_X(x) = P_{U_1}(u'_x) + P_{U_1}(\mathcal{L}''(\mathcal{U}_1))P_X(x) \tag{5.38}$$

where $u'_x = \{u' \in \mathcal{L}'(\mathcal{U}_1) : x = f(u')\}$. This recursive expression can be understood as follows. The first component corresponds to the case when we parse a binary string using $\mathcal{U}_1$ such that it is in $\mathcal{L}'(\mathcal{U}_1)$ and it mapped to x. Second component of the expression corresponds to the case when we parse a binary string using $\mathcal{U}_1$ such that it is in $\mathcal{L}''(\mathcal{U}_1)$ and we simply discard this parsed string and start parsing again the rest of the input sequence using $\mathcal{U}_1$. The reason we can discard this parsed string is that the value of the function $f(u)$ in independent of this segment of the concatenation since it is from $\mathcal{L}''(\mathcal{U}_1)$. Moreover when we start parsing again the rest of the input sequence it is independent of the already parsed segment hence the probability of x occuring given we are again at the root of $\mathcal{U}_1$ in parsing process is the same as before parsing the previous segment from $\mathcal{L}''(\mathcal{U}_1)$. and the probability of ending up again at the root of $\mathcal{U}_1$ without generating any ouput symbol is precisely $P_{U_1}(\mathcal{L}''(\mathcal{U}_1))$. Note that we only need to store $\mathcal{U}_1$ for this encoder instead of $\mathcal{U}$ for the parsing and mapping process. Hence the implementation complexity(in terms of the memory requirements for the tree $\mathcal{U}$) of this encoder is the same as of a v2f encoder with maximum delay constraint.

Using argument similar to the ones presented in (5.33), the fact that $\mathcal{L}'(\mathcal{U}_1)$ and $\mathcal{L}''(\mathcal{U}_1)$ form a partition of $\mathcal{L}(\mathcal{U}_1)$ and $\bigcup_{x \in \mathcal{L}(\mathcal{X})} u'_x = \mathcal{L}'(\mathcal{U}_1)$ we have for any $x \in \mathcal{L}(\mathcal{X})$

$$P_{U_1}(u'_x) = \frac{M_x}{2^m} \tag{5.39}$$

$$P_{U_1}(\mathcal{L}''(\mathcal{U}_1)) = \frac{2^m - M}{2^m} \tag{5.40}$$

hence for any $x \in \mathcal{L}(\mathcal{X})$

$$P_X(x) = \frac{M_x}{2^m} + \frac{2^m - M}{2^m} P_X(x) \tag{5.41}$$

$$= \frac{M_x}{M} \tag{5.42}$$

where $\sum_{x \in \mathcal{L}(\mathcal{X})} M_x = M$.Hence for FSG encoder the approximating distribution P_X is M-type for some value of M between 1 and 2^m.

In the next step to characterize the set of approximating distributions that can be generated by such encoders we show that for any M-type distribution P where $1 \leq M \leq 2^m$, we can design an encoder satisfying the constraints mentioned in this subsection such that $P_X = P$. For any M-type distribution P use the following algorithm to construct such an encoder.

Algorithm 5.

$\mathcal{L}(\mathcal{U}_1) = \{0,1\}^m$

$\mathcal{L}'(\mathcal{U}_1)$ = Set of m bit binary representations of first M non-negative integers.

$i = 1$

repeat while $i \leq |\operatorname{supp}(P)|$

1. $n_i = P(i) \cdot M$

2. Form a set of m bit binary representations of all integers from $\sum_{k=1}^{i-1} n_m$ to $\sum_{k=1}^{i} n_m - 1$ and label this set as u'_i. Assign all the strings in u'_i to symbol i

4. $i \leftarrow i + 1$

5. end

Proposition 15. *A probability distribution P can be generated by the v2f encoder discussed in this section iff P is M-type where $1 \leq M \leq 2^m$.*

Now that we have characterized the kind of distributions that can be generated using an FSG encoder, we present the procedure for designing the desired FSG encoder for approximating a target product distribution. The first step is to find the K-type informational divergence optimal quantization of P_Y^n using Algo. 2.7.2.2 where K can take any value between 1 and 2^m. Denote this optimal quantization by P_X and let M be its type. Now use the following algorithm

Algorithm 6.

---

1. Define $P'_X = \left[\frac{M}{2^l} P_X \quad \frac{2^l - M}{2^l}\right] \qquad 2^{l-1} < M \leq 2^l,\ l \in Z$

2. Construct Knuth encoder for P'_X. Denote the binary tree for this encoder by $\mathcal{U}_1$.

3. Label the set of u corresponding to $P'_X(i)$ as u'_i for $1 \leq i \leq |\operatorname{supp}(P_X)|$. Denote the set of u corresponding to last entry of P'_X, i.e., $\frac{2^l - M}{2^l}$ as $\mathcal{L}''(\mathcal{U}_1)$.

Since P'_X is 2^l-type so the Knuth encoder for it will be such that no $u \in \mathcal{L}(\mathcal{U}_1)$ has $\ell(u) > l$ hence this encoder satisfies the constraints mentioned earlier.

Remark 14. *In the process of designing the encoder in this section we have also characterized the set of distributions corresponding to a special case of FSGs. This question was asked in [1] for general FSG.*

Remark 15. *The encoder proposed in this subsection and knuth encoder may not be identical even for M-type target distribution for* $1 \leq M \leq 2^m$. *Consider the example where* $m = 3$ *and* $P_Y = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{3}{5} \end{bmatrix}$ *and* $n = 1$. *In this case knuth encoder and encoder defined in this section will not be identical.*

Rate, Distortion: We know that the set of approximating distributions we minimize the informational divergence over in this subsection for a given m and n is a superset of the set of distributions we optimize over in previous subsection. Hence from this observation we can conclude that $\forall \epsilon > 0$, if $\frac{m}{n} > \mathbb{H}(P_Y) + \epsilon$ then $\mathbb{D}(P_X \| P_Y^{\mathcal{X}}) \overset{n\to\infty}{\longrightarrow} 0$. Moreover we know that the type of approximating distribution is between 1 and 2^m. Hence $R_{hv} \leq \frac{m}{n}$ which shows the achievability for $(R_{hv}, \mathbb{D}(P_X \| P_Y^{\mathcal{X}}))$.
Since the constructed encoder is a knuth encoder for P'_X yielding $\mathcal{U}_1$, we have [1]

$$\mathbb{H}(P'_X) \leq \mathbb{E}[\ell(U_1)] \leq \mathbb{H}(P'_X) + 2 \tag{5.43}$$

where

$$\mathbb{H}(P'_X) = (1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))\, \mathbb{H}(P_X) + \mathbb{H}_b(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1))) \tag{5.44}$$

Moreover based on the recursive construction of $\mathcal{U}$ from $\mathcal{U}_1$ we can conclude that

$$\mathbb{E}[\ell(U)] = \frac{\mathbb{E}[L(U_1)]}{1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1))} \tag{5.45}$$

where $(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1))) = \frac{M}{2^l}$. Combining (5.43), (5.44), (5.45) and the fact that $\mathcal{X} = \mathcal{Y}^n$ we have

$$\frac{\mathbb{H}(P_X)}{n} + \frac{\mathbb{H}_b(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))}{n(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))} \leq R \leq \frac{\mathbb{H}(P_X)}{n} + \frac{\mathbb{H}_b(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))}{n(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))} + \frac{2}{n(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))} \tag{5.46}$$

To make the bounds independent of $P_{U_1}(\mathcal{L}''(\mathcal{U}_1))$ we evaluate the lower and upper bounds on $P_{U_1}(\mathcal{L}''(\mathcal{U}_1))$ and $\frac{\mathbb{H}_b(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))}{(1 - P_{U_1}(\mathcal{L}''(\mathcal{U}_1)))}$. $P_{U_1}(\mathcal{L}''(\mathcal{U}_1))$ is lower bounded by 0 and achieves this lower bound for the cases when the P_X is 2^l-type. It is upper bounded by $\frac{1}{2}$ since $2^{l-1} \leq M \leq 2^l$. This upperbound can also be implied based on scaling property for Alg. 2.7.2.2 in Remark. 1. Hence $0 \leq \frac{\mathbb{H}_b(1 - P_U(u_e))}{(1 - P_U(u_e))} < 2$. Using these bounds

$$\frac{\mathbb{H}(P_X)}{n} \leq R < \frac{\mathbb{H}(P_X)}{n} + \frac{6}{n} \tag{5.47}$$

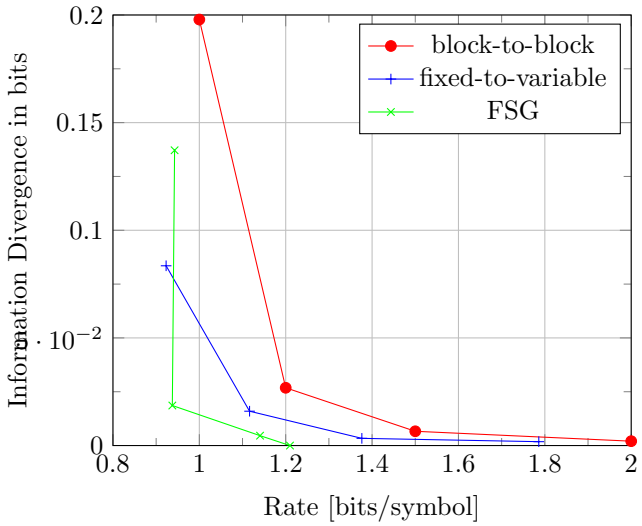

Figure 5.3: Comparison of rate vs distortion performance of zero error Distribution Matching techniques for target distribution $P_Y(0) = 1 - P_Y(1) = \frac{1}{3}$.

Combining Entropy-Divergence Theorem (Prop. 1) and the fact that for such encoders $\mathbb{D}(P_X \| P_Y^{\mathcal{X}}) \overset{n\to\infty}{\longrightarrow} 0$ as long as $\frac{m}{n} > \mathbb{H}(P_Y)$ we conclude that $R_k \to \mathbb{H}(P_Y)$ as $n \to \infty$ and $\frac{m}{n} > \mathbb{H}(P_Y)$ proving achievability for R_k.

Remark 16. ***Tighter bound for expected length of Knuth encoder:*** *For an arbitrary target distribution P_Y one cannot compute the $\mathbb{E}[\ell(U)]$ for a Knuth encoder but for 2^m-type distribution it is possible to compute $\mathbb{E}[\ell(U)]$ for a Knuth encoder. Using* (5.45) *we can compute the $\mathbb{E}[\ell(U)]$ since U_1 corresponds to a knuth encoder with 2^m-type target distribution. For those rational distributions over finite alphabets where* (5.45) *yields a value lower than $\mathbb{H}(P_Y)+2$ we obtain a tighter bound on the $\mathbb{E}[\ell(U)]$ for Knuth encoder.An example is $P_Y = \frac{1}{31}\begin{bmatrix}3 & 4 & 4 & 2 & 8 & 2 & 8\end{bmatrix}$ where $\mathbb{H}(P_Y) = 2.60$ and $\mathbb{E}[\ell(U)] = 2.90$ for the FSG encoder providing a much tighter upper bound then $\mathbb{H}(P_Y) + 2$.*

5.7 Comparison

To conclude this chapter we present a comparison between the optimal block to block encoder, fixed to variable length encoder and FSG encoder. We have chosen FSG encoder among the two variable to fixed length encoders since it provides better distortion performance for the same maximum length constraint than the other encoder. The rate distortion performance of the three encoders is presented in Fig 5.3. We clearly see that the variable length encoders outperform the optimal block to block resolution encoder.

6 Conclusions

Simulation of DMS P_Y from a DMS P_B, alternatively known as approximate random number generation from a fair coin, has recently found its application to many new problems including information theoretic secrecy[26], coordination[27] and probabilistic shaping for reliable data transmission[6] in addition to the classical applications such as system simulations. In this thesis we have developed a unified framework to study the following two important variants of approximate random number generation, both inspired from the above mentioned applications

- Distribution Matching
- Resolution Coding for Target Distributions.

In Chap. 3 we have developed Entropy-Divergence Theorem which bounds the difference between the entropy rate of a variable length code and a memoryless source in terms of normalized informational divergence. We have used it several times in the following chapters for proving converses and showing achievability. Although this bound can be used to show asymptotic results, this appears to be loose for various codes developed in finite length regime. Hence an interesting direction of research is to find a stronger result which gives tighter bounds for entropy.
In Chap. 4 we have studied distribution matching in detail. A converse establishing an upper bound on the achievable rates is proven. The upper bound is the entropy of the target distribution. Achievability for this rate upper bound is shown using one-to-one b2b matcher based on typicality. We proceed with proposing optimal one-to-one b2b matcher, one-to-one f2v matcher and ϵ-error b2b matcher. Asymptotic achievability has been proven for all of the matchers. Moreover, we have shown asymptotic achievability for the v2f matcher proposed in [6]. Some of the open problems for future research in distribution matching regime are

- The f2v matcher we have proposed is distortion optimal under certain restrictions on target distribution or the sufficiently large input block length. One open problem is to find the distortion optimal fixed-to-variable matcher for any input block length and any target distribution.

- Develop a rate-distortion theory for distribution matching problem.
- Develop a theory parallel to the one developed in Chap. 4 for distribution matching at the output of a channel. This finds its applications for probabilistic shaping for DMCs as well as for channel coding for DMCs.

In Chap. 5 we discussed resolution coding for target distributions. We have generalised the converse presented in [3] in the special case of identity channel and memoryless stationary random process to make it applicable to variable length encoders. In order to develop a unified approach from both information theoretic and algorithmic complexity perspective we have discussed the rate and distortion measures used in both [1] and [3]. We have first established relationship among these measures and then we have developed optimal b2b encoders, f2v encoders and v2f encoders and have proven achievability for all of these codes. Some of the directions for future research are

- Application of the codes developed in this thesis for secrecy and coordination problems.
- Developing similar codes to achieve the bounds shown in [3] for approximation of output statistics.
- Characterization of probability distributions corresponding to different classes of Finite State Generators. This question was raised in [1] and was partially answered. We have also characterized the set of distributions corresponding to a special class of FSGs in Sec. 5.6.2. Characterizing the set of distributions corresponding to other classes of Finite State Generators is an interesting problem both in context of approximate random number generation and finite automata.
- In [2] authors have proposed an algorithm for exact random number generation similar to arithmetic coding. Constructing practical encoders based on arithmetic for approximate random number generation is an interesting research problem.

Another important research problem, which has been partially studied up in [2] in the context of exact random number generation, is to develop the theory parallel to the one in this thesis for simulating any DMS from any other DMS, instead of DMS P_B as was the case in this thesis.
Although the discussion in this thesis was focussed on approximate random number generation, in the process of constructing different encoders we have solved various discrete optimization problem that may have their own importance other than for approximate random nunmber generation. These include

- Minimization of $\mathbb{D}(P_U \| P_Y^{\mathcal{L}(\mathcal{X})})$ over complete trees $\mathcal{X}$.
- Minimization of $\mathbb{D}(P \| P_Y^{\mathcal{L}(\mathcal{X})})$ over P such that P can be only be M-type for $1 \leq M \leq 2^m$.
- Characterization of the P_B alphabet induced set of probability distributions over leaves of trees corresponding to a special case of binary FSGs.
- A tighter upper bound on the expected length of Knuth encoder for certain rational distributions over finite alphabets.

In a nutshell, approximate random number generation and approximation of output statistics have recently found interesting application in various classical information theory problems. The construction of practical codes for these problems and development of a unified framework for their analysis is still largely an open problem and this thesis is an effort in this direction.

Bibliography

[1] D. Knuth and A. Yao, *The complexity of nonuniform random number generation.* New York: Academic Press, 1976, pp. 357–428.

[2] T. S. Han and M. Hoshi, "Interval algorithm for random number generation," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 599–611, 1997.

[3] T. S. Han and S. Verdu, "Approximation theory of output statistics," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 752–772, 1993.

[4] Y. Steinberg and S. Verdu, "Simulation of random processes and rate-distortion theory," *IEEE Trans. Inf. Theory*, vol. 42, no. 1, pp. 63–86, 1996.

[5] J. Hou and G. Kramer, "Informational divergence approximations to product distributions," 2013. [Online]. Available: http://arxiv.org/abs/1302.0215

[6] G. Böcherer, "Capacity-achieving probabilistic shaping for noisy and noiseless channels," Ph.D. dissertation, RWTH Aachen University, 2012. [Online]. Available: http://www.georg-boecherer.de/capacityAchievingShaping.pdf

[7] A. Wyner, "The common information of two dependent random variables," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 163–179, 1975.

[8] J. L. Massey, "Applied digital information theory I," lecture notes, ETH Zurich. [Online]. Available: http://www.isiweb.ee.ethz.ch/archive/massey_scr/adit1.pdf

[9] G. Böcherer and B. C. Geiger, "Optimal M-type quantizations of distributions," Jul. 2013. [Online]. Available: http://arxiv.org/abs/1307.6843

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.

[11] R. A. Rueppel and J. L. Massey, "Leaf-average node-sum interchanges in rooted trees with applications," in *Communications and Cryptography: Two sides of One Tapestry*, R. E. Blahut, D. J. Costello Jr., U. Maurer, and T. Mittelholzer, Eds. Kluwer Academic Publishers, 1994.

[12] G. Böcherer and R. A. Amjad, "Informational divergence and entropy rate on rooted trees with probabilities," 2013, submitted to Int. Zurich Seminar Commun. [Online]. Available: http://arxiv.org/abs/1310.2882

[13] J. L. Massey, "An information-theoretic approach to algorithms," in *The Impact of Processing Techniques on Communications*. Springer, 1985, pp. 3–22. [Online]. Available: http://www.isiweb.ee.ethz.ch/archive/massey_pub/pdf/BI309.pdf

[14] R. A. Amjad and G. Böcherer, "Fixed-to-variable length distribution matching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2013, pp. 1511–1515.

[15] G. Böcherer and R. A. Amjad, "Block-to-block distribution matching," 2013. [Online]. Available: http://arxiv.org/abs/1302.1020

[16] T. S. Han, *Information Spectrum Methods in Information Theory*. Springer Verlag, 2003.

[17] G. Kramer, "Information theory," lecture notes TU Munich, edition WS 2012/2013.

[18] G. Böcherer and R. Mathar, "Matching dyadic distributions to channels," in *Proc. Data Compression Conf.*, 2011, pp. 23–32. [Online]. Available: http://arxiv.org/abs/1009.3751

[19] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[20] B. Varn, "Optimal variable length codes (arbitrary symbol cost and equal code word probability)," *Inf. Contr.*, vol. 19, no. 4, pp. 289 – 301, 1971.

[21] G. Böcherer, "Block-to-block distribution matching," *Joint Conference on Coding and Communications*, 2013, poster.

[22] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[23] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.

[24] G. Böcherer and R. A. Amjad, "Fixed-to-variable length resolution coding for target distributions," *IEEE Inf. Theory Workshop (ITW)*, 2013.

[25] G. Kramer, "Multi-user information theory," lecture notes TU Munich, edition SS 2012.

[26] S. S. Y. Liang, H. V. Poor, *Information Theoretic Security*, 1st ed. now publisher, 2009.

[27] P. Cuff, H. Permuter, and T. Cover, "Coordination capacity," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4181–4206, 2010.