

2016

Mutual dimension, data processing inequalities, and randomness

Adam Case

Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Case, Adam, "Mutual dimension, data processing inequalities, and randomness" (2016). *Graduate Theses and Dissertations*. 15674.
<https://lib.dr.iastate.edu/etd/15674>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Mutual dimension, data processing inequalities, and randomness

by

Adam Thomas Case

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:

Jack H. Lutz, Major Professor

Pavan Audri

Clifford Bergman

Timothy H. McNicholl

Giora Slutzki

Iowa State University

Ames, Iowa

2016

Copyright © Adam Thomas Case, 2016. All rights reserved.

DEDICATION

I dedicate this dissertation to my family and friends. Your love and support means so much to me.

TABLE OF CONTENTS

| | |
|---|-----|
| DEDICATION | ii |
| ACKNOWLEDGMENTS | vi |
| ABSTRACT | vii |
| CHAPTER 1. INTRODUCTION | 1 |
| 1.1 Notions of Dimension | 2 |
| 1.2 Betting Strategies and Constructive Dimension | 2 |
| 1.3 Algorithmic Information Theory and Constructive Dimension | 7 |
| 1.4 Applications of Effective Dimension | 9 |
| 1.4.1 Computational Complexity | 10 |
| 1.4.2 Fractal Geometry | 10 |
| 1.4.3 Other Applications | 11 |
| 1.5 Overview of Chapter 2 | 11 |
| 1.6 Overview of Chapter 3 | 13 |
| 1.7 Overview of Chapter 4 | 16 |
| 1.8 Overview of Chapter 5 | 19 |
| CHAPTER 2. KOLMOGOROV COMPLEXITY AND MUTUAL IN- | |
| FORMATION IN EUCLIDEAN SPACE | 23 |
| 2.1 Preliminaries | 24 |
| 2.2 Layered Disjoint Systems and a Coding Theorem | 25 |
| 2.3 Counting K-minimizers within Blocks of a LDS | 28 |

| | | |
|---|--|----|
| 2.4 | Counting K -minimizers within Cubes and Balls | 29 |
| 2.5 | Upper Bounds on $K_r(x)$ and $K_{r+s}(x)$ | 33 |
| 2.6 | Algorithmic Mutual Information in Euclidean Space | 36 |
| 2.7 | Relating $I_r(x : y)$ and $J_r(x : y)$ | 38 |
| 2.8 | Properties of $I_r(x : y)$ | 42 |
| CHAPTER 3. MUTUAL DIMENSION AND DATA PROCESSING | | |
| | INEQUALITIES | 45 |
| 3.1 | Mutual Dimension in Euclidean Space | 46 |
| 3.2 | Computable Functions in Euclidean Space | 48 |
| 3.3 | Data Processing Inequalities for Points in Euclidean Space | 48 |
| 3.4 | Reverse Data Processing Inequalities | 52 |
| 3.5 | Data Processing Applications | 59 |
| CHAPTER 4. BOUNDED TURING REDUCTIONS AND DATA PRO- | | |
| | CESSING INEQUALITIES FOR SEQUENCES | 61 |
| 4.1 | Notation | 62 |
| 4.2 | Relating the Kolmogorov Complexities of Sequences to the Kolmogorov Complexities of Reals | 62 |
| 4.3 | Relating the Dimensions of Sequences to the Dimensions of Reals | 66 |
| 4.4 | Relating the Mutual Information between Sequences to the Mutual Infor- mation between Reals | 67 |
| 4.5 | Relating the Mutual Dimensions between Sequences to the Mutual Di- mensions between Reals | 70 |
| 4.6 | Properties of Mutual Dimensions between Sequences | 71 |
| 4.7 | Turing Reductions and Functionals | 72 |
| 4.8 | Turing Functionals with Bounded Use and Data Processing Inequalities | 73 |
| 4.9 | Turing Functionals with Bounded Yield and Reverse Data Processing In- equalities | 78 |

| | |
|--|------------|
| CHAPTER 5. COUPLED RANDOMNESS | 83 |
| 5.1 Probability Measures on Alphabets and Sequences | 83 |
| 5.2 Coupled Probability Measures | 84 |
| 5.3 Coupled Random Sequences | 86 |
| 5.4 Shannon Entropy Characterizations of the Dimensions of Random Sequences | 87 |
| 5.5 Shannon Mutual Information Characterizations of the Mutual Dimensions of Coupled Random Sequences | 88 |
| 5.6 The Mutual Dimension between Independently Random Sequences | 90 |
| 5.7 Dependent Sequences with Zero Mutual Dimension | 90 |
| 5.8 Billingsley Dimension | 92 |
| 5.9 Billingsley Mutual Dimension | 93 |
| 5.10 A Mutual Divergence Formula | 94 |
| 5.11 Achieving Mutual Normalizability | 95 |
| BIBLIOGRAPHY | 101 |

ACKNOWLEDGMENTS

I would like to thank my mom, dad, sister, and brother for their loving support while I've been away in Iowa working on my Ph.D. I am especially thankful to my parents for instilling in me a healthy work ethic.

I would also like to thank Jack Lutz for his willingness to take me on as one of his Ph.D. students. I can honestly say that it has been an honor (and a lot of fun!) working with him over the past six and a half years. Much of what I have learned as both a researcher and a teacher has come from him.

I have had the pleasure of working with several excellent researchers. Joshua Case, Debasis Mandal, Titus Klinge, Don Stull, Xiang Huang, Samuel Eillis, Don Nye, Timothy McNicholl, Jim Lathrop, Pavan Aduri, and Giora Slutzki have all been a great source of intellectual stimulation. I've enjoyed the conversations we've had over the years, and I hope that we will have many more!

Finally, I'd like to thank the faculty in the Mathematics and Computer Science Department at the University of Maine at Farmington for giving me an appreciation for my field of study. Their enthusiasm and dedication to teaching is inspiring to me.

This research was supported in part by National Science Foundation Grants 1247051 and 1545028.

ABSTRACT

This dissertation makes progress in the area of constructive dimension, an effectivization of classical Hausdorff dimension. Using constructive dimension, one may assign a non-zero number to the dimension of individual sequences and individual points in Euclidean space. The primary objective of this dissertation is to develop a framework for *mutual dimension*, i.e., the *density* of algorithmic mutual information between two infinite objects, that has similar properties as those of classical Shannon mutual information.

Chapter 1 presents a brief history of the development of constructive dimension along with its relationships to algorithmic information theory, algorithmic randomness, and classical Hausdorff dimension. Some applications of this field are discussed and an overview of each subsequent chapter is provided.

Chapter 2 defines and analyzes the mutual algorithmic information between two points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ at a given precision $r \in \mathbb{N}$. In fact, we describe two plausible definitions for this quantity, $I_r(x : y)$ and $J_r(x : y)$, and show that they are closely related. In order to do this, we prove and make use of a generalization of Levin's coding theorem.

Chapter 3 defines the lower and upper mutual dimensions between two points in Euclidean space and presents results on its basic properties. A large portion of this chapter is dedicated to studying data processing inequalities for points in Euclidean space. Generally speaking, a data processing inequality says that the amount of information between two objects cannot be significantly increased when one of the objects is processed by a particular type of transformation. We show that it is possible to derive several kinds of

data processing inequalities for points in Euclidean space depending on the continuity properties of the computable transformation that is used.

Chapter 4 focuses on extending mutual dimension to sequences over an arbitrary alphabet. First, we prove that the mutual dimension between two sequences is equal to the mutual dimension between the sequences' real representations. Using this result, we show that the lower and upper mutual dimensions between sequences have nice properties. We also provide an analysis of data processing inequalities for sequences where transformations are represented by Turing functionals whose use and yield are bounded by computable functions.

Chapter 5 relates mutual dimension to the study of algorithmic randomness. Specifically, we show that a particular class of *coupled random sequences*, i.e., sequences generated by independent tosses of coins whose biases may or may not be correlated, can be characterized by classical Shannon mutual information. We also prove that any two sequences that are independently random with respect to computable probability measures have zero mutual dimension and that the converse of this statement is not true. We conclude this chapter with some initial investigations on Billingsley mutual dimension, i.e., mutual dimension with respect to probability measures, and prove the existence of a mutual divergence formula.

CHAPTER 1. INTRODUCTION

In this introductory chapter, we provide an overview of the history of constructive dimension. A brief intuition of the dimension of geometric objects in Euclidean space is discussed along with Hausdorff's notion of dimension. We then explore formal betting strategies for sequences, known as *martingales*, and describe how to characterize randomness using constructive martingales.

In the early 2000's, Lutz used *s*-gales, a more general betting strategy than a martingale, to characterize classical Hausdorff dimension. We discuss how Lutz developed constructive dimension, an effectivization of classical Hausdorff dimension, by applying certain computability restrictions to *s*-gales. Perhaps one of the most surprising consequences of this effectivization is the ability to assign a non-zero dimension to an individual sequence or an individual point in Euclidean space. This notion of dimension has been shown to be geometrically meaningful and is useful in several areas of mathematics and computer science.

An overview of algorithmic information theory is also given and includes a brief discussion on the history and basic definitions of Kolmogorov complexity and algorithmic probability. We also discuss how constructive dimension can be characterized in terms of Kolmogorov complexity, and we end the section by providing an overview of each chapter of this dissertation.

This chapter is a joint work with Jack H. Lutz and some portions of it have appeared in [10, 9, 8].

1.1 Notions of Dimension

Until the end of the 19th century, mathematicians typically associated the *dimension* of a geometric object with a non-negative integer $\{0, 1, 2, \dots\}$. It was generally accepted that objects such as points were 0-dimensional (since they “lacked width and length”) and curves were considered 1-dimensional (since they “lacked width”). However, in 1890, Giuseppe Peano constructed the first ever space-filling curve that is a continuous function defined on the unit interval and goes through every point in the unit square. This was considered a major accomplishment that went against commonly held intuitions about geometry and dimension. Is this space-filling curve considered 1-dimensional or 2-dimensional?

In 1918, Felix Hausdorff developed a rigorous notion of dimension that not only assigns a non-negative integer to simple geometric objects such as lines, squares, cubes, etc., but also assigns a value greater than $n - 1$ and less than n to some of the more complex objects in \mathbb{R}^n . Indeed, it has been shown that the Cantor set $\mathcal{C} \subseteq \mathbb{R}$ has Hausdorff dimension $dim_H(\mathcal{C}) = \frac{\ln(2)}{\ln(3)} \approx 0.631$ and the Sierpinski triangle $\mathcal{S} \subseteq \mathbb{R}^2$ has Hausdorff dimension $dim_H(\mathcal{S}) = \frac{\log(3)}{\log(2)} \approx 1.585$. These strange, yet beautiful, sets were first referred to as “fractals” by Benoît Mandelbrot due to their fractional dimension [49, 50, 19].

1.2 Betting Strategies and Constructive Dimension

In this section, we discuss some of the basic ideas from algorithmic randomness, how Lutz originally defined constructive dimension using betting strategies, and how constructive dimension is related to classical Hausdorff dimension.

For the following definitions, we write $\{0, 1\}^*$ for the set of all finite binary strings and \mathbf{C} for the set of all infinite binary sequences. We denote the *empty string* by λ .

A well-known formalization for betting on the bits of a sequence is to define a *martingale*, i.e., a function $d : \{0, 1\}^* \rightarrow [0, \infty)$ such that

$$d(w) = \frac{d(w0) + d(w1)}{2},$$

where $d(\lambda) = 1$. Intuitively, we can think of $d(w)$ as the amount of money we have after betting on the bits of w , where we start with 1 dollar before we begin betting. If, for some $w \in \{0, 1\}^*$, $d(w) = 0$ (i.e., we run out of money after betting on w), then we cannot win any more money betting on any string that extends w , e.g., $d(w0) = d(w1) = 0$. Notice that bets are “fair” in the sense that the amount of money we have after betting on the bits of w is equal to the average amount of money we have after betting on $w0$ and $w1$.

For every sequence $S \in \mathbf{C}$, a martingale d *succeeds* on S if

$$\limsup_{n \rightarrow \infty} d(S \upharpoonright n) = \infty, \quad (1.2.1)$$

where $S \upharpoonright n$ denotes the first n bits of S , i.e., we win an infinite amount of money when using d as a strategy for betting on the bits of S . It is easy to see that, for any sequence $S \in \mathbf{C}$, we can define the following martingale,

$$d(w) = \begin{cases} 2^{|w|} & \text{if } w \sqsubseteq S \\ 0 & \text{if } w \not\sqsubseteq S \end{cases}, \quad (1.2.2)$$

for all $w \in \{0, 1\}^*$, that will succeed on S . However, d is not necessarily an effective strategy since it may not be possible to compute an approximation of $d(w)$, let alone compute its actual value.

A function $f : \{0, 1\}^* \rightarrow [0, \infty)$ is called *lower semicomputable* if there exists a computable function $\hat{f} : \{0, 1\}^* \times \mathbb{N} \rightarrow \mathbb{Q}$ such that, for all $(x, n) \in \{0, 1\}^* \times \mathbb{N}$,

$$\hat{f}(x, n) \leq \hat{f}(x, n + 1),$$

and, for all $x \in \{0, 1\}^*$, $\lim_{n \rightarrow \infty} \hat{f}(x, n) = f(x)$. A function is *constructive* if it is lower semicomputable.

We call a sequence *random* if there exists no constructive martingale that succeeds on it. The intuition here is that, if there exists no constructive martingale that succeeds on a sequence, then the bits of this sequence are not predictable. Claus-Peter Schnorr [58] showed that this notion of randomness is equivalent to *Martin-Löf randomness* [51].

In developing constructive dimension, Lutz defined a new kind of betting strategy that generalizes the notion of a martingale. Let $s \geq 0$. An *s-supergale* is a function $d : \{0, 1\}^* \rightarrow [0, \infty)$ such that

$$d(w) \geq \frac{d(w0) + d(w1)}{2^s}.$$

We say that d is an *s-gale* if the above inequality holds with equality. Observe that a 1-gale is a martingale and, therefore, is fair. However, if $s < 1$, then the betting environment becomes more hostile (less fair) since $d(w0) + d(w1)$ becomes smaller. On the other hand, if $s > 1$, then the betting environment becomes less hostile (more fair) since $d(w0) + d(w1)$ becomes larger.

As with martingales, an *s-supergale* d *succeeds* on a sequence S if (1.2.1) holds.

The set of all binary sequences \mathbf{C} is a metric space with the metric

$$d(S, T) = \begin{cases} 2^{-r} & \text{if } S \neq T \text{ and } r = \min\{n \in \mathbb{N} \mid S[n] \neq T[n]\} \\ 0 & \text{if } S = T \end{cases}.$$

Therefore, we may reason about the Hausdorff dimension of any subset $X \subseteq \mathbf{C}$. Lutz proved the following characterization of Hausdorff dimension in [43].

Theorem 1.2.1. *For all $X \subseteq \mathbf{C}$,*

$$\dim_H(X) = \inf\{s \geq 0 \mid \text{there exists an } s\text{-gale that succeeds on every } S \in X\}.$$

The classical Hausdorff dimension $\dim_H(X)$ of a set $X \subseteq \mathbb{R}^n$ is defined in terms of set covers, i.e., unions of balls that contain X . Another notion of dimension, called *packing dimension*, was developed by Claude Tricot in 1982 [32]. The packing dimension of a set $X \subseteq \mathbb{R}^n$, denoted $\dim_P(X)$, is similar to Hausdorff dimension but is defined using

unions of disjoint balls whose centers must be contained within X [19]. There are many sets X where $\dim_H(X)$ and $\dim_P(X)$ differ, but in general

$$\dim_H(X) \leq \dim_P(X).$$

In 2007, Athreya, et al. defined that an s -gale *succeeds strongly* on a sequence $S \in \mathbf{C}$ if

$$\liminf_{n \rightarrow \infty} d(S \upharpoonright n) = \infty$$

and proved that the packing dimension of a set $X \subseteq \mathbf{C}$ may be characterized as follows using s -gales [2].

Theorem 1.2.2. *For all $X \in \mathbf{C}$,*

$$\dim_P(X) = \inf\{s \geq 0 \mid \text{there exists an } s\text{-gale that strongly succeeds on every } S \in X\}.$$

Intuitively, we can think of $\dim_H(X)$ and $\dim_P(X)$ as the most hostile betting environment $s \geq 0$ such that, for every $S \in X$, an infinite amount of money can be won by betting on the bits of S using some s -gale betting strategy. Since these are characterizations of classical Hausdorff and packing dimension, there exist no computability restrictions on the gales.

In [44], Lutz effectivized Hausdorff dimension by defining the *constructive dimension* of a set $X \subseteq \mathbf{C}$ using constructive s -supergales, and John Hitchcock proved that constructive dimension may be equivalently defined by restricting Lutz's definition to s -gales [28]. In [2], Athreya et al. effectivized packing dimension by defining the *constructive strong dimension* of a set $X \subseteq \mathbf{C}$.

Definition. The *constructive dimension* and *constructive strong dimension* of a set $X \subseteq \mathbf{C}$ are

$$cdim(X) = \inf\{s \geq 0 \mid \text{there exists a constructive } s\text{-gale that succeeds on every } S \in X\}$$

and

$$cDim(X)$$

$= \inf\{s \geq 0 \mid \text{there exists a constructive } s\text{-gale that succeeds strongly on every } S \in X\}$,
respectively.

It is easy to see that, for all $S \in \mathbf{C}$, $dim_H(\{S\}) = dim_P(\{S\}) = 0$ since there exists an s -gale that is similar to the martingale defined in (1.2.2). However, using constructive dimension, we may (perhaps surprisingly) assign a non-zero value to the dimension of an *individual* sequence.

Definition. The *lower* and *upper dimensions* of $S \in \mathbf{C}$ are

$$dim(S) = cdim(\{S\})$$

and

$$Dim(S) = cDim(\{S\}),$$

respectively.

In [44], Lutz demonstrates that the dimension of a sequence has nice properties. For example, for every sequence $S \in \mathbf{C}$, $dim(S) \in [0, 1]$, and, for every $\alpha \in [0, 1]$, there exists an uncountable number of sequences S such that $dim(S) = \alpha$. Also, it is easy to verify that if $R \in \mathbf{C}$ is random, then $dim(R) = 1$. On the other hand, if S is computable, then $dim(S) = 0$. While constructive martingales have provided a means of reasoning about the structure of random sequences, constructive s -gales provides a means of analyzing the structure of sequences that are not necessarily random. $Dim(S)$ has similar properties, and, as with Hausdorff and packing dimension,

$$dim(S) \leq Dim(S),$$

for all sequences S .

It is also possible to characterize the constructive dimension and constructive strong dimension of a set $X \subseteq \mathbf{C}$ using the lower and upper dimensions of the individual sequences within X . The following theorem was proven in [44, 2].

Theorem 1.2.3. For all $X \in \mathbf{C}$,

$$cdim(X) = \sup_{S \in X} dim(S)$$

and

$$cDim(X) = \sup_{S \in X} Dim(S).$$

In 2005, Hitchcock proved a pointwise characterization for the Hausdorff dimension of a union of Π_1^0 sets [29].

Theorem 1.2.4. If $X \subseteq \mathbf{C}$ is a union of Π_1^0 sets, then

$$dim_H(X) = \sup_{S \in X} dim(S).$$

Chris Conidis proved that the packing dimension of a union $X \subseteq \mathbf{C}$ of Π_1^0 sets cannot be characterized by the supremum of the upper dimensions of the individual sequences in X [12].

1.3 Algorithmic Information Theory and Constructive

Dimension

Betting strategies such as martingales are one way of reasoning about the randomness of objects. However, other paradigms have been developed that provide equivalent ways of thinking about randomness. One of these paradigms is based on the *minimum-length description* of a string and is often referred to as *Kolmogorov complexity* due to Andrey Kolmogorov's work in this area. It is worth noting that the foundations for this field were originally developed between 1960 and 1964 by Ray Solomonoff [65, 66, 67] and then independently discovered by Kolmogorov in 1965 [36] and Gregory Chaitin in 1969 [11]. In this section, we provide a basic overview of algorithmic probability and *prefix* Kolmogorov complexity, which will be expanded upon in Chapter 2. For an in-depth

analysis of this subject, we refer the reader to the books by Ming Li and Paul Vitányi [42], André Nies [54], and Rodney Downey and Denis Hirschfeldt [17].

A set of strings $S \subseteq \Sigma^*$ is *prefix-free* if no string in S is a prefix of any other string in S . A *self-delimiting Turing machine* is a Turing machine M whose domain (i.e., the set of all strings that M halts on) is a prefix-free set. It is well-known that there exist *universal* self-delimiting Turing machines. For the rest of this dissertation, we call a self-delimiting Turing machine simply a Turing machine, and we let U be some fixed universal self-delimiting Turing machine.

Definition. The *Kolmogorov complexity* of a string $w \in \{0, 1\}^*$ is

$$K(w) = \min\{|\pi| \mid \pi \in \{0, 1\}^* \text{ and } U(\pi) = w\}.$$

The intuition of the above definition is that any string $\pi \in \{0, 1\}^*$ such that $K(w) = |\pi|$ and $U(\pi) = w$ is one (of perhaps many) of the most *compressed* representations of w that is *decompressible* by U . The length of this minimum-length description $|\pi|$ is the quantity of information content in w .

Strings that are *incompressible*, i.e., strings w such that $K(w) \geq |w|$, are considered *random*. In 1973, both Leonid Levin [38] and Schnorr [60] used Kolmogorov complexity to characterize random sequences.

Theorem 1.3.1. *A sequence $S \in \mathbf{C}$ is random if and only if there exists a constant $c \in \mathbb{N}$ such that, for all $n \in \mathbb{N}$,*

$$K(S \upharpoonright n) \geq n - c.$$

Solomonoff defined the *universal a priori probability* of a string w as

$$\mathbf{m}(w) = \sum_{U(\pi)=w} 2^{-|\pi|},$$

i.e., $\mathbf{m}(w)$ is the probability that the universal Turing machine U outputs w when U is given an input π such that each bit of π is produced by a fair coin toss. Solomonoff

used \mathbf{m} as a tool in the development of his theory of inductive inference and has several interesting applications in the field of artificial intelligence.

In 1974, Leonid Levin proved his coding theorem, which relates the universal a priori probability of a string to its Kolmogorov complexity [38, 39].

Theorem 1.3.2. *For all strings w ,*

$$K(w) = \log \frac{1}{\mathbf{m}(w)} + O(1).$$

Theorem 2.2.1 of this dissertation generalizes Levin's coding theorem.

Kolmogorov complexity has also been useful in the theory of constructive dimension. In 2002, Elvira Mayordomo proved that the lower dimension of a binary sequence can be characterized in terms of Kolmogorov complexity [52], and, in 2007, Athreya et al. proved a similar result for the upper dimension of a binary sequence [2].

Theorem 1.3.3. *For all sequences $S \in \mathbf{C}$,*

$$\dim(S) = \liminf_{n \rightarrow \infty} \frac{K(S \upharpoonright n)}{n}$$

and

$$\text{Dim}(S) = \limsup_{n \rightarrow \infty} \frac{K(S \upharpoonright n)}{n}.$$

With these characterizations, we can view the dimension of a sequence S as its *density* of algorithmic information. This way of describing constructive dimension is useful to this dissertation since we define the mutual dimension between two sequences, i.e., the density of *shared* algorithmic information between two sequences, using a similar characterization.

1.4 Applications of Effective Dimension

There are many useful applications of effective dimension. In this section, we provide a brief overview of some of these applications, which is not meant to be exhaustive. We

encourage the reader to refer to the survey papers [45, 31] for a more in-depth discussion on this topic.

1.4.1 Computational Complexity

One of the first applications of effective dimension was in the field of computational complexity. Lutz considered time and space bounded s -gales in order to discuss the structure of sets inside of complexity classes and proved several results on frequency sets in E and circuit-size complexity in ESPACE [43]. Other interesting results that use dimension in complexity classes include several dimension zero-one laws. For example, Lance Fortnow et al. proved that the strong dimension of E in ESPACE is either 0 or 1 [20] and Moser proved that either the dimension of BPP in EXP is 0 or $BPP = EXP$ [53]. Hitchcock and Gavaldà et al. have also related resource-bounded dimension to computational learning [30, 23].

1.4.2 Fractal Geometry

Researchers have used Kolmogorov complexity to develop a notion of the dimension of an *individual* point in \mathbb{R}^n [47]. The *Kolmogorov complexity* of a point $x \in \mathbb{R}^n$ at *precision* $r \in \mathbb{N}$ is

$$K_r(x) = \min\{K(q) \mid q \in B_{2^{-r}}(x) \cap \mathbb{Q}^n\}, \quad (1.4.1)$$

where $B_{2^{-r}}(x)$ is the open ball of radius 2^{-r} centered at x . Here, $K(q)$ is the length of a shortest program that outputs a *binary encoding* of q . The idea of this definition is to assign $K_r(x)$ to be $K(q)$, for some representative $q \in B_{2^{-r}}(x) \cap \mathbb{Q}^n$. One might ask, what is considered an appropriate representative? Several of these rationals within $B_{2^{-r}}(x)$ will include a large amount of spurious information. Indeed, any finite-length message can be encoded into one of these rational points. By assigning $K_r(x)$ to be the minimum $K(q)$, we ensure that q only has information that its proximity to x forces it to have.

We may also assign a dimension to an individual point $x \in \mathbb{R}^n$ that is not necessarily zero.

Definition. The *lower* and *upper dimensions* of a point $x \in \mathbb{R}^n$ are

$$\dim(x) = \liminf_{r \rightarrow \infty} \frac{K_r(x)}{r}.$$

and

$$\text{Dim}(x) = \limsup_{r \rightarrow \infty} \frac{K_r(x)}{r}.$$

Using the constructive dimensions of points, it is possible to perform a point-wise analysis of self-similar fractals. For example, in [47], Lutz and Mayordomo characterized the dimensions of individual points of computably self-similar fractals. In [15], Dougherty et al. analyzed the constructive dimension of points that are the result of a translation of a point in the Cantor set by a random real. In [25], Gu et al. considered random subfractals $S \subseteq F$ of self-similar fractals $F \subseteq \mathbb{R}^n$ and studied their dimension spectra, i.e., the set of all dimensions of points within S .

1.4.3 Other Applications

The dimensions of points have also been used to study connectivity properties [48, 68], rectifiability of curves [56, 24], and Brownian motion [34].

1.5 Overview of Chapter 2

Claude E. Shannon defined the *entropy* of a random variable X with outcomes $\{x_1, x_2, \dots, x_n\}$ to be

$$\mathcal{H}(X) = \sum_{i=1}^n p(x_i) \log_2 \frac{1}{p(x_i)}.$$

Intuitively, $\mathcal{H}(X)$ is the expected number of bits of information revealed by the outcome of X . We can define the *mutual information* between random variables X and Y to be

$$I(X; Y) = \mathcal{H}(Y) - \mathcal{H}(Y | X),$$

where $\mathcal{H}(Y | X)$ is the *conditional entropy* of Y given X . We can think of $I(X; Y)$ as the shared information between X and Y [14]. Analogously, we define the *algorithmic mutual information* $I(u : w)$ between two strings $u \in \Sigma^*$ and $w \in \Sigma^*$ to be

$$I(u : w) = K(w) - K(w | u),$$

where

$$K(w | u) = \min\{|\pi| \mid \pi \in \{0, 1\}^* \text{ and } U(\pi, u) = w\}$$

is the *conditional* Kolmogorov complexity of w given u . Like Shannon mutual information, $I(u : w)$ represents the quantity of information that both u and w share. In fact, it has been shown that, under modest assumptions, if x and y are drawn from probability spaces X and Y of strings, respectively, then the expected value of $I(x : y)$ is very close to $I(X; Y)$ [42]. In this sense, algorithmic mutual information is a refinement of Shannon mutual information.

One way of measuring the algorithmic mutual information between $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ is by considering the mutual information between the prefixes of the binary expansions of their individual components. However, Turing's correction to his 1973 paper [69] indicates that there exist very simple functions (e.g., addition) that are not computable when reals are represented by their binary expansions. Since Chapter 3 addresses how computable functions process points in Euclidean space, we use a different method for measuring the mutual information between points that is based on rational approximations of reals.

We define the algorithmic mutual information $I_r(x : y)$ between $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ at a given precision $r \in \mathbb{N}$ to be the minimum $I(q : p)$ such that $q \in B_{2^{-r}}(x) \cap \mathbb{Q}^n$ and $p \in B_{2^{-r}}(y) \cap \mathbb{Q}^t$. The intuition for this definition is similar to that of $K_r(x)$ in (1.4.1). Any $q \in B_{2^{-r}}(x)$ and $p \in B_{2^{-r}}(y)$ such that $I(q : p) = I_r(x : y)$ will only share the information that their proximities to x and y force them to share.

Another way of thinking about the algorithmic mutual information between two points is by considering the mutual information between the rationals $q \in B_{2^{-r}}(x)$ and $p \in B_{2^{-r}}(y)$ that have minimum Kolmogorov complexity within their respective balls. More precisely, let $J_r(x : y) = I(q : p)$ such that $q \in B_{2^{-r}}(x) \cap \mathbb{Q}^n$, $p \in B_{2^{-r}}(y) \cap \mathbb{Q}^t$, $K(q) = K_r(x)$, and $K(p) = K_r(y)$. Intuitively, this seems to be a good alternative definition for the algorithmic mutual information between two points at precision r . In fact, Theorem 2.7.7 says that, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$, $I_r(x : y) = J_r(x : y) + o(r)$.

It can be useful to consider both I_r and J_r when reasoning about mutual information between points. For example, we use Theorem 2.7.7 to prove the following properties of $I_r(x : y)$.

1. $I_r(x : y) = K_r(x) + K_r(y) - K_r(x, y) + o(r)$.
2. $I_r(x : y) \leq \min\{K_r(x), K_r(y)\} + o(r)$.
3. If x and y are independently random, then $I_r(x : y) = o(r)$.
4. $I_r(x : y) = I_r(y : x) + o(r)$.

In order to prove this relationship between $I_r(x : y)$ and $J_r(x : y)$, we must first establish an upper bound on the number of rational points $q \in \mathbb{Q}^n$ of minimum Kolmogorov complexity within an arbitrary ball of radius 2^{-r} (Theorem 2.4.4). We prove this upper bound by making use of a generalization of Levin's coding theorem [38, 39], which is one of the main theorems of Chapter 1 (Theorem 2.2.1).

1.6 Overview of Chapter 3

In Chapter 2, we explore the algorithmic mutual information between points in Euclidean space. The first section of Chapter 3 defines the mutual dimension between two point and analyzes its basic properties. The *lower* and *upper mutual dimensions* between

$x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ are

$$mdim(x : y) = \liminf_{r \rightarrow \infty} \frac{I_r(x : y)}{r}$$

and

$$Mdim(x : y) = \limsup_{r \rightarrow \infty} \frac{I_r(x : y)}{r},$$

respectively. Our first theorem of this chapter, Theorem 3.1.1, describes the basic properties of mutual dimension. For all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$, the following hold.

1. $dim(x) + dim(y) - Dim(x, y) \leq mdim(x : y) \leq Dim(x) + Dim(y) - Dim(x, y)$.
2. $dim(x) + dim(y) - dim(x, y) \leq Mdim(x : y) \leq Dim(x) + Dim(y) - dim(x, y)$.
3. $mdim(x : y) \leq \min\{dim(x), dim(y)\}$, $Mdim(x : y) \leq \min\{Dim(x), Dim(y)\}$.
4. $0 \leq mdim(x : y) \leq Mdim(x : y) \leq \min\{n, t\}$.
5. If x and y are independently random, then $Mdim(x : y) = 0$.
6. $mdim(x : y) = mdim(y : x)$, $Mdim(x : y) = Mdim(y : x)$.

(Note that property 5 will be discussed in detail in Chapter 5.) The above properties for mutual dimension include all but one of the desiderata (e.g., see Bell [3]) for any satisfactory notion of mutual information. Section 3.3 is dedicated to investigating the most important desideratum, the *data processing inequality* for both $mdim$ and $Mdim$.

Intuitively, a data processing inequality states that the amount of shared information between two objects will never significantly increase after one of these objects is processed by a particular kind of function. Various subfields of information theory have developed their own kinds of data processing inequalities. For example, in classical Shannon information theory [14], for all probability spaces X , Y , and Z and all functions $f : X \rightarrow Z$,

$$I(f(X); Y) \leq I(X; Y).$$

In algorithmic information theory, if $f : \Sigma^* \rightarrow \Sigma^*$ is a computable function, then there exists a constant $c \in \mathbb{N}$ such that, for all strings $x, y \in \Sigma^*$,

$$I(f(x) : y) \leq I(x : y) + c. \quad (1.6.1)$$

In order to investigate data processing inequalities for points in \mathbb{R}^n , we must be able to reason about the computability of functions in Euclidean space. The framework we use for this is taken from the field of computable analysis as found in the works of Ko [35], Weihrauch [73], and Braverman and Cook [6]. An *oracle* for a point $x \in \mathbb{R}^n$ is a computable function $g_x : \mathbb{N} \rightarrow \mathbb{Q}^n$ such that, for all $n \in \mathbb{N}$, $|g_x(n) - x| \leq 2^{-n}$. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is *computable* if there exists an oracle machine M such that, for every oracle g_x for $x \in \mathbb{R}^n$ and every $n \in \mathbb{N}$, $|M^{g_x}(n) - f(x)| \leq 2^{-n}$, i.e., M^{g_x} is an oracle for $f(x)$.

Given (1.6.1), it might seem reasonable to conjecture that if $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is computable, then

$$mdim(f(x) : y) \leq mdim(x : y) \quad \text{and} \quad Mdim(f(x) : y) \leq Mdim(x : y), \quad (1.6.2)$$

for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, but this does not hold in general. For example, it has been shown that there exist functions $f : \mathbb{R} \rightarrow \mathbb{R}^2$ that are both computable and space-filling (e.g., $[0, 1]^2 \subseteq \text{range}(f)$) [13]. Therefore, if $x \in \mathbb{R}$ such that $dim(f(x)) = 2$, then

$$\begin{aligned} mdim(f(x) : f(x)) &= dim(f(x)) \\ &= 2 \\ &> 1 \\ &\geq Dim(x) \\ &\geq Mdim(x : y). \end{aligned}$$

The problem here is that f is extremely sensitive to its input, which allows it to compress a great deal of “sparse” high-precision information about its input x into “dense” lower-precision information about its output $f(x)$. To avoid these excessively sensitive

functions, we require that our computable functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ be *Lipschitz*, i.e., there exists a constant $c > 0$ such that, for all $x, y \in \mathbb{R}^n$, $|f(x) - f(y)| \leq c \cdot |x - y|$. The main theorem of this chapter, which we refer to as *the data processing inequality*, says that, for all computable Lipschitz functions $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ and all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, (1.6.2) holds.

To prove the data processing inequality, we first prove a more general result called the *modulus processing lemma*. Using this lemma, we show the existence of inequalities similar to that of (1.6.2), whose functions f are not necessarily Lipschitz. For example, we show that, if a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is *Hölder with exponent* $\alpha \in (0, 1]$ (i.e., there exists a constant $c > 0$ such that, for all $x, y \in \mathbb{R}^n$, $|f(x) - f(y)| \leq c \cdot |x - y|^\alpha$), then

$$mdim(f(x) : y) \leq \frac{1}{\alpha} mdim(x : y) \quad \text{and} \quad Mdim(f(x) : y) \leq \frac{1}{\alpha} Mdim(x : y),$$

In Section 3.4, we derive *reverse* data processing inequalities, for example, giving conditions under which $mdim(x : y) \leq mdim(f(x) : y)$. In Section 3.5, we use data processing inequalities and their reverses to explore conditions under which computable functions on Euclidean space preserve, approximately preserve, or otherwise transform mutual dimensions between points.

1.7 Overview of Chapter 4

In this chapter, we extend the notion of mutual dimension to sequences over an arbitrary alphabet Σ . Formally, the *lower* and *upper* mutual dimensions between sequences $S \in \Sigma^\infty$ and $T \in \Sigma^\infty$ are defined by

$$mdim(S : T) = \liminf_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{|u| \log |\Sigma|}$$

and

$$Mdim(S : T) = \limsup_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{|u| \log |\Sigma|},$$

respectively. The first objective of this chapter is to prove the basic properties for the lower and upper mutual dimensions between sequences, which are similar to the properties for the lower and upper mutual dimensions between points in Euclidean space. We accomplish this goal by relating the mutual dimensions between sequences to the mutual dimensions between the sequences' real representations. The primary objective of this chapter is to analyze how the lower and upper mutual dimensions between two sequences change when one of the sequences is transformed by a Turing functional.

A reduction can be described in several ways. Generally speaking, a problem A reduces to a problem B if A is solvable when assuming that B is solvable. In computability theory, Turing reductions are used to discuss the idea of relative computability. Formally, a sequence S is *Turing reducible* to a sequence T if there exists an oracle machine that computes S when T is written on the oracle tape. We often refer to oracle machines as Turing functionals, which have been studied in detail by Rogers [57] and Soare [63, 64]. When a Turing functional Φ^S runs on a particular input, it is allowed to query the oracle S at any time. The *use* of a Turing functional is the largest position of the oracle tape that is queried during the computation of Φ^S on input n . We will be primarily concerned with Turing functionals whose use is bounded by a computable function.

Downey, Hirshfeldt, and LaForte first defined *sw-reducibility* (strong weak truth table reducibility) as a Turing reduction whose use is bounded by $n + c$ where $n \in \mathbb{N}$ is the input and c is a constant [16]. The authors showed that, for all sequences S and T , if T is sw-reducible to S , then, for all $n \in \mathbb{N}$,

$$K(T \upharpoonright n) \leq K(S \upharpoonright n) + O(1).$$

An sw-reduction is now referred to as a *computable Lipschitz reduction* (*cl-reduction*) because all Turing functionals whose use function is bounded by $n + c$ can be viewed as an effective Lipschitz continuous function [41, 40].

In Section 4.8, we discuss data processing inequalities for sequences, where transformations are represented by Turing functionals with bounded use. Our main result of

this section says that, for all sequences $X, Y, Z \in \Sigma^\infty$, if Z is cl-reducible to X , then

$$mdim(Z : Y) \leq mdim(X : Y)$$

and

$$Mdim(Z : Y) \leq Mdim(X : Y).$$

We also show that, for all $\alpha \geq 1$, if Z is reducible to X via a functional Φ whose use is bounded by $\lceil \alpha(n + c) \rceil$, for all inputs $n \in \mathbb{N}$, then

$$mdim(Z : Y) \leq \alpha \cdot mdim(X : Y)$$

and

$$Mdim(Z : Y) \leq \alpha \cdot Mdim(X : Y).$$

We then provide weaker versions of the above inequalities stated in terms of the Turing functionals themselves.

In section 4, we explore *reverse data processing inequalities* for sequences, i.e., data processing inequalities where the transformation may significantly *increase* the amount of shared information between two objects. Unlike the data processing inequalities described above, we cannot derive reverse data processing inequalities by restricting how much of the oracle a Turing functional accesses. Instead, we place restrictions on the lengths of the strings that a Turing functional outputs.

In [22], Gács analyzed the lengths of the outputs of monotonic operators, which are also used to describe Turing reductions. Similarly, we are interested in examining the lengths of the strings output by a Turing functional equipped with a finite oracle. We define the *yield* of a Turing functional Φ^S with access to at most $n \in \mathbb{N}$ bits of the oracle S , denoted $\phi_{yield}^S(n)$, to be the smallest input $m \in \mathbb{N}$ such that $\Phi^{S \upharpoonright n}(m) \uparrow$.

We say that a sequence T is *uniquely yield bounded reducible* (*uyb-reducible*) to a sequence S if there exists a Turing functional Φ such that,

1. if the first $\phi_{yield}^S(n)$ symbols of Φ^S is a prefix of Φ^T , then the first n symbols of S is a prefix of T , and
2. $\phi_{yield}^S(n)$ is bounded by a computable function.

Our main result of this section says that, for all sequences $X, Y, Z \in \Sigma^\infty$, if Z is uyb-reducible to X via a functional Φ such that $\phi_{yield}^X(n) \leq n + c$, for some constant $c \in \mathbb{N}$, then

$$mdim(X : Y) \leq mdim(Z : Y)$$

and

$$Mdim(X : Y) \leq Mdim(Z : Y).$$

We also show that, for all $\alpha \geq 1$, if Z is uyb-reducible to X via a functional Φ such that $\phi_{yield}^X(n) \leq \lceil \alpha(n + c) \rceil$, for all inputs $n \in \mathbb{N}$, then

$$mdim(X : Y) \leq \alpha \cdot mdim(Z : Y)$$

and

$$Mdim(X : Y) \leq \alpha \cdot Mdim(Z : Y).$$

1.8 Overview of Chapter 5

With the exception of property 5 in Theorem 3.1.1, we have not yet discussed the relationships between algorithmic randomness and mutual dimension. In this chapter, we investigate the mutual dimension between *coupled random* sequences, produce some results on algorithmic independence, and explore a notion of constructive mutual Billingsley dimension.

In Section 1.2, we defined a sequence $R \in \Sigma^\infty$ to be random if there exists no constructive martingale that succeeds on R . Intuitively, a random sequence $R \in \{0, 1\}^\infty$ is one whose bits are generated by the outcomes of infinitely many tosses of a fair coin.

We may also define other types of random sequences by using biased coins. For example, let $\vec{\alpha} = (\alpha^{(1)}, \alpha^{(2)}, \dots)$ be a sequence of probability measures on $\{0, 1\}$. A sequence $R \in \{0, 1\}^\infty$ is *random* with respect to $\vec{\alpha}$ if, for each $i \in \mathbb{N}$, the i th bit of R is the outcome of an independent $\alpha^{(i)}$ -biased coin toss.

In [44], Lutz showed that, for any sequence $R \in \{0, 1\}^\infty$ that is random with respect to a computable sequence $\vec{\alpha}$ of probability measures on $\{0, 1\}$ that converges to a probability measure α on $\{0, 1\}$, then

$$\dim(R) = \mathcal{H}(\alpha). \quad (1.8.1)$$

This theorem can be thought of as an algorithmic extension of a classical theorem of Eggleston [18, 5]

When discussing coupled randomness, we must consider probability measures on $\{0, 1\} \times \{0, 1\}$. Let $\vec{\alpha} = \{\alpha^{(1)}, \alpha^{(2)}, \dots\}$ be a sequence of probability measures on $\{0, 1\} \times \{0, 1\}$. We say that a pair of sequences $(R_1, R_2) \in \{0, 1\}^\infty \times \{0, 1\}^\infty$ is *coupled random* with respect to $\vec{\alpha}$ if R_1 is random with respect to $\vec{\alpha}_1 = (\alpha_1^{(1)}, \alpha_1^{(2)}, \dots)$ and R_2 is random with respect to $\vec{\alpha}_2 = (\alpha_2^{(1)}, \alpha_2^{(2)}, \dots)$, where $\alpha_1^{(i)}$ is the first marginal probability measure of $\alpha^{(i)}$ and $\alpha_2^{(i)}$ is the second marginal probability measure on $\alpha^{(i)}$. Intuitively, the i th bit of R_1 is the outcome of an $\alpha_1^{(i)}$ -biased coin toss and the i th bit of R_2 is the outcome of an $\alpha_2^{(i)}$ -biased coin toss. Notice that the i th bit of R_1 is generated *independently* of the $i + 1$ th bit of R_1 , but the i th bits of R_1 and R_2 may be *correlated* since $\alpha_1^{(i)}$ and $\alpha_2^{(i)}$ may be *dependent* probability measures. We make this definition precise and extend it to sequences over an arbitrary alphabet Σ in Chapter 5.

The main theorem of this chapter, Theorem 5.5, states that, for every pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is coupled random with respect to a computable sequence $\vec{\alpha}$ of probability measures on $\Sigma \times \Sigma$ that converges to a probability measure α on $\Sigma \times \Sigma$,

$$mdim(R_1 : R_2) = Mdim(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\log |\Sigma|}.$$

This theorem can be regarded as a “mutual version” of (1.8.1) that has also been generalized for random sequences over an arbitrary alphabet Σ . We also show that $Mdim(R_1 : R_2) = 0$ is a necessary, but not sufficient condition for two random sequences R_1 and R_2 to be independently random.

A probability measure on Σ^∞ is a function $\beta : \Sigma^* \rightarrow [0, 1]$ such that

1. $\nu(\lambda) = 1$, where λ is the empty string and
2. for every $w \in \Sigma^*$, $\beta(w) = \sum_{a \in \Sigma} \beta(wa)$.

Intuitively, $\beta(w)$ is the probability that $w \sqsubseteq S$ (w is a prefix of S) when $S \in \Sigma^\infty$ is “chosen according to” the probability measure β . A probability measure β on Σ^∞ is *strongly positive* if there exists a $\delta > 0$ such that, for all $w \in \Sigma^*$ and $a \in \Sigma$, $\beta(wa) > \delta\beta(w)$.

In 1960 Billingsley investigated generalizations of Hausdorff dimension in which the dimension itself is defined “through the lens of” a given probability measure [4, 7]. Lutz and Mayordomo developed the *lower* and *upper effective Billingsley dimensions* $dim^\beta(S)$ and $Dim^\beta(S)$ defined by

$$dim^\beta(S) = \liminf_{w \rightarrow S} \frac{K(w)}{\ell_\beta(w)}$$

and

$$Dim^\beta(S) = \limsup_{w \rightarrow S} \frac{K(w)}{\ell_\beta(w)},$$

where β is a strongly positive probability measure on Σ^∞ and

$$\ell_\beta(w) = \sum_{i=0}^{|w|-1} \log \frac{1}{\beta(w[i])}$$

is the *Shannon self-information* of $w \in \Sigma$ with respect to β , i.e., the number of bits of information contained in w , where each symbol $a \in \Sigma$ is given a weight $\beta(a)$. It is an easy observation that if μ is the uniform probability measure on Σ^∞ , then $dim^\mu(S) = dim(S)$ and $Dim^\mu(S) = Dim(S)$. In a sense, the probability measure β in the above definitions acts as a “standard for randomness” since sequences $S \in \Sigma^\infty$ that are random with

respect to β are the only kind of random sequences where $\dim^\beta(S) = \text{Dim}^\beta(S) = 1$. These effective Billingsley dimensions have been useful in the algorithmic information theory of self-similar fractals [47, 25].

Our final objective is to investigate “Billingsley generalizations” $\text{mdim}^\nu(S : T)$ and $\text{Mdim}^\nu(S : T)$ of $\text{mdim}(S : T)$ and $\text{Mdim}(S : T)$, where ν is a probability measure on $\Sigma^\infty \times \Sigma^\infty$. These turn out to make sense only when S and T are *mutually normalizable*, which means that the normalizations implicit in the fact that these dimensions are *densities* of shared information are the same for S as for T . We prove that, when mutual normalizability is satisfied, the Billingsley mutual dimensions $\text{mdim}^\nu(S : T)$ and $\text{Mdim}^\nu(S : T)$ are well behaved. We also identify a sufficient condition for mutual normalizability, make some preliminary observations on when it holds, and prove a divergence formula, analogous to a theorem of [46], for computing the values of the Billingsley mutual dimensions in many cases.

CHAPTER 2. KOLMOGOROV COMPLEXITY AND MUTUAL INFORMATION IN EUCLIDEAN SPACE

In this chapter, we develop the underlying framework required to discuss mutual dimension in Euclidean space. We define a layered disjoint system (LDS), which allows us to view certain discrete spaces in terms of “layers” that are partitioned into “blocks.” Our first result generalizes Levin’s Coding Theorem ([38, 39]) and relates the Kolmogorov complexity of a block within an LDS to its universal a priori probability. Using this result, we establish an upper bound on the number of strings of minimal Kolmogorov complexity within a particular block.

The Kolmogorov complexity of a point x in \mathbb{R}^n at precision $r \in \mathbb{N}$, denoted by $K_r(x)$, is defined as the minimum Kolmogorov complexity of a rational point within $B_r(x)$, which is the open ball of radius r centered at x . A rational point $q \in \mathbb{Q}^n \cap B_r(x)$ such that $K(q) = K_r(x)$ is called a K -minimizer of $B_r(x)$. We prove upper bounds on the number of K -minimizers within a ball of radius $r \in \mathbb{N}$ and the values of $K_r(x)$ and $K_{r+s}(x)$, for some $s \in \mathbb{N}$.

Finally, we define the mutual information between two points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ at precision $r \in \mathbb{N}$, denoted by $I_r(x : y)$, to be the minimum mutual information between a rational point in $B_r(x)$ and a rational point in $B_r(y)$. We prove that $I_r(x : y) = J_r(x : y) + o(r)$ as $r \rightarrow \infty$, where $J_r(x : y)$ is the minimum mutual information between a K -minimizer of $B_r(x)$ and a K -minimizer of $B_r(y)$. We conclude this chapter with a discussion on the basic properties of $I_r(x : y)$.

This chapter is a joint work with Jack H. Lutz and appeared in [10].

2.1 Preliminaries

We write \mathbb{Z} for the set of integers, \mathbb{N} for the set of non-negative integers, \mathbb{Q} for the set of rationals, \mathbb{R} for the set of reals, and \mathbb{R}^n for the set of all n -vectors (x_1, x_2, \dots, x_n) such that each $x_i \in \mathbb{R}$. Our logarithms are in base 2. We denote the cardinality of a set A , the length of a string $s \in \{0, 1\}^*$, and the distance between two points $x, y \in \mathbb{R}^n$ (using the Euclidean metric) by $|A|$, $|s|$, and $|x - y|$ respectively. We also denote the i^{th} string in $\{0, 1\}^*$ by s_i .

Our use of Turing machines is strictly limited to self-delimiting (or prefix) machines. Because of this, we refer to a self-delimiting Turing machine simply as a Turing machine. We refer the reader to Li and Vitanyi [42] for a detailed explanation of how self-delimiting Turing machines work.

The (*conditional*) *Kolmogorov complexity* of a string $x \in \{0, 1\}^*$ given a string $y \in \{0, 1\}^*$ with respect to a Turing machine M is

$$K_M(x | y) = \min\{|\pi| \mid \pi \in \{0, 1\}^* \text{ and } M(\pi, y) = x\}.$$

The *Kolmogorov complexity* of x with respect to M is $K_M(x) = K_M(x | \lambda)$, where λ is the *empty string*. A Turing machine M' is *optimal* if, for every Turing machine M , there is a constant $c_M \in \mathbb{N}$ such that, for all $x \in \{0, 1\}^*$,

$$K_{M'}(x) \leq K_M(x) + c_M.$$

We call c_M an *optimality constant* for M . It is well-known that every universal Turing machine is optimal [42]. Following standard practice, we fix a universal, hence optimal, Turing machine U ; we omit it from the notation, writing $K(x) = K_U(x)$ and $K(x | y) = K_U(x | y)$; and we call these the *Kolmogorov complexity* of x and the (*conditional*) *Kolmogorov complexity* of x given y , respectively.

The *joint Kolmogorov complexity* of two strings $x, y \in \{0, 1\}^*$ is

$$K(x, y) = K(\langle x, y \rangle),$$

where $\langle \cdot, \cdot \rangle$ is some standard pairing function for encoding two strings. Gács [21] proved the useful identity

$$K(x, y) = K(x) + K(y | \langle x, K(x) \rangle) + O(1). \quad (2.1.1)$$

The *universal a priori probability* of a set $S \subseteq \{0, 1\}^*$ is

$$\mathbf{m}(S) = \sum_{U(\pi) \in S} 2^{-|\pi|}.$$

Since we are using self-delimiting machines, the Kraft inequality tells us that $\mathbf{m}(\{0, 1\}^*) \leq 1$. The *universal a priori probability* of a string $x \in \{0, 1\}^*$ is $\mathbf{m}(x) = \mathbf{m}(\{x\})$. For $r \in \mathbb{N}$, we write $K(r)$ for $K(s_r)$ and $\mathbf{m}(r)$ for $\mathbf{m}(s_r)$. It is well known that there is a constant $c_0 \in \mathbb{N}$ such that $K(x) \leq |x| + 2 \log(1 + |x|) + c_0$, and hence $K(r) \leq \log(1 + r) + 2 \log(1 + \log(1 + r)) + c_0$, hold for all $x \in \{0, 1\}^*$ and $r \in \mathbb{N}$.

Levin's coding lemma plays an important role in section 2.2.

Lemma 2.1.1 (coding lemma [38, 39]). *If $A \subseteq \{0, 1\}^* \times \mathbb{N}$ is computably enumerable and satisfies $\sum_{(x,l) \in A} 2^{-l} \leq 1$, then there is a Turing machine M such that, for each $(x, l) \in A$, there is a string $\pi \in \{0, 1\}^l$ satisfying $M(\pi) = x$.*

2.2 Layered Disjoint Systems and a Coding Theorem

We begin by developing some elements of the fine-scale geometry of algorithmic information in Euclidean space. In this context it is convenient to regard the Kolmogorov complexity of a set of strings to be the number of bits required to specify *some* element of the set.

Definition (Shen and Vereshchagin [62]). The *Kolmogorov complexity* of a set $S \subseteq \{0, 1\}^*$ is

$$K(S) = \min\{K(x) \mid x \in S\}.$$

Note that $S \subseteq T$ implies $K(S) \geq K(T)$. Intuitively, small sets may require “higher resolution” than large sets.

We need a generalization of Levin’s coding theorem [38, 39] that is applicable to certain systems of disjoint sets.

Notation. Let $B \subseteq \mathbb{N} \times \mathbb{N} \times \{0, 1\}^*$ and $r, s \in \mathbb{N}$.

1. The (r, t) -block of B is the set $B_{r,t} = \{x \in \{0, 1\}^* \mid (r, t, x) \in B\}$.
2. The r^{th} layer of B is the sequence $B_r = (B_{r,t} \mid t \in \mathbb{N})$.

Definition. A *layered disjoint system* (LDS) is a set $B \subseteq \mathbb{N} \times \mathbb{N} \times \{0, 1\}^*$ such that, for all $r, s, t \in \mathbb{N}$,

$$s \neq t \Rightarrow B_{r,s} \cap B_{r,t} = \emptyset.$$

Note that this definition only requires the sets within each layer of B to be disjoint.

Theorem 2.2.1 (LDS coding theorem). *For every computably enumerable layered disjoint system B there is a constant $c_B \in \mathbb{N}$ such that, for all $r, t \in \mathbb{N}$,*

$$K(B_{r,t}) \leq \log \frac{1}{\mathbf{m}(B_{r,t})} + K(r) + c_B.$$

Proof. Assume the hypothesis, and fix a computable enumeration of B . For each $r, t \in \mathbb{N}$ such that $B_{r,t} \neq \emptyset$, let $x_{r,t}$ be the first element of $B_{r,t}$ to appear in this enumeration. Let A be the set of all ordered pairs $(x_{r,t}, j + k + 2)$ such that $r, t, j, k \in \mathbb{N}$, $B_{r,t} \neq \emptyset$, $k \geq K(r)$, and $\mathbf{m}(B_{r,t}) \geq 2^{-j}$. It is clear that A is computably enumerable.

For each $r, t \in \mathbb{N}$, let

$$j_{r,t} = \min\{j \in \mathbb{N} \mid \mathbf{m}(B_{r,t}) > 2^{-j}\},$$

noting that $j_{r,t} = \infty$ if $B_{r,t} = \emptyset$. For all $r, t \in \mathbb{N}$ such that $B_{r,t} \neq \emptyset$, we have

$$\begin{aligned}
\sum_{\substack{l \in \mathbb{N} \\ (x_{r,t}, l) \in A}} 2^{-l} &= \sum_{j=j_{r,t}}^{\infty} \sum_{k=K(r)}^{\infty} 2^{-(j+k+2)} \\
&= \sum_{k=K(r)}^{\infty} 2^{-(k+1)} \sum_{j=j_{r,t}}^{\infty} 2^{-(j+1)} \\
&= 2^{-K(r)} 2^{-j_{r,t}} \\
&< 2^{-K(r)} \mathbf{m}(B_{r,t}).
\end{aligned}$$

Since the sets in each layer B_r of B are disjoint, it follows that

$$\begin{aligned}
\sum_{(x,l) \in A} 2^{-l} &\leq \sum_{r=0}^{\infty} \sum_{t=0}^{\infty} 2^{-K(r)} \mathbf{m}(B_{r,t}) \\
&= \sum_{r=0}^{\infty} 2^{-K(r)} \sum_{t=0}^{\infty} \mathbf{m}(B_{r,t}) \\
&= \sum_{r=0}^{\infty} 2^{-K(r)} \mathbf{m}\left(\bigcup_{t=0}^{\infty} B_{r,t}\right) \\
&\leq \sum_{r=0}^{\infty} 2^{-K(r)} \mathbf{m}(\{0, 1\}^*) \\
&\leq \sum_{r=0}^{\infty} 2^{-K(r)} \\
&\leq \sum_{r=0}^{\infty} \mathbf{m}(r) \\
&= \mathbf{m}(\{0, 1\}^*) \\
&\leq 1.
\end{aligned}$$

We have now shown that the set A satisfies the hypothesis of Lemma 2.1.1. Let M be a Turing machine for A as in that lemma, and let $c_B = c_M + 3$, where c_M is an optimality constant for M . To see that c_B affirms the theorem, let $r, t \in \mathbb{N}$ be such that $B_{r,t} \neq \emptyset$. (The theorem is trivial if $B_{r,t} = \emptyset$, since the right-hand side is infinite.) Then $(x_{r,t}, j_{r,t} + K(r) + 2) \in A$, so there is a program $\pi \in \{0, 1\}^{j_{r,t} + K(r) + 2}$ such that

$M(\pi) = x_{r,t}$. We thus have

$$\begin{aligned}
K(B_{r,t}) &\leq K(x_{r,t}) \\
&\leq K_M(x_{r,t}) + c_M \\
&\leq |\pi| + c_M \\
&= j_{r,t} + K(r) + 2 + c_M \\
&= \lfloor \log \frac{1}{\mathbf{m}}(B_{r,t}) \rfloor + 1 + K(r) + 2 + c_M \\
&\leq \log \frac{1}{\mathbf{m}(B_{r,t})} + K(r) + c_B. \quad \square
\end{aligned}$$

Note that Levin's coding theorem [38, 39], the nontrivial part of which says that $K(x) \leq \log \frac{1}{\mathbf{m}(x)} + O(1)$, is the special case $B_{r,t} = \{s_t\}$ of the LDS coding theorem.

2.3 Counting K -minimizers within Blocks of a LDS

Our next objective is to use the LDS coding theorem to obtain useful bounds on the number of times that the value $K(S)$ is attained or approximated.

Definition. Let $S \subseteq \{0, 1\}^*$ and $d \in \mathbb{N}$.

1. A d -approximate K -minimizer of S is a string $x \in S$ for which $K(x) \leq K(S) + d$.
2. A K -minimizer of S is a 0-approximate K -minimizer of S .

We use the LDS coding theorem to prove the following.

Theorem 2.3.1. *For every computably enumerable layered disjoint system B there is a constant $c_B \in \mathbb{N}$ such that, for all $r, t, d \in \mathbb{N}$, the block $B_{r,t}$ has at most $2^{d+K(r)+c_B}$ d -approximate K -minimizers.*

Proof. Let B be a computably enumerable LDS, and let c_B be as in the LDS coding theorem. Let $r, t, d \in \mathbb{N}$, and let N be the number of d -approximate K -minimizers of the block $B_{r,t}$. Then

$$\mathbf{m}(B_{r,t}) \geq N \cdot 2^{-(K(B_{r,t})+d)},$$

so the LDS coding theorem tells us that

$$\begin{aligned} K(B_{r,t}) &\leq \log \frac{1}{N \cdot 2^{-(K(B_{r,t})+d)}} + K(r) + c_B \\ &= K(B_{r,t}) + d - \log N + K(r) + c_B. \end{aligned}$$

This implies that

$$\log N \leq d + K(r) + c_B,$$

whence

$$N \leq 2^{d+K(r)+c_B}. \quad \square$$

2.4 Counting K -minimizers within Cubes and Balls

We now lift our terminology and notation to Euclidean space \mathbb{R}^n . In this context, a layered disjoint system is a set $B \subseteq \mathbb{N} \times \mathbb{N} \times \mathbb{R}^n$ such that, for all $r, s, t \in \mathbb{N}$,

$$s \neq t \Rightarrow B_{r,s} \cap B_{r,t} = \emptyset.$$

We lift our Kolmogorov complexity notation and terminology to \mathbb{R}^n in two steps:

1. Lifting to \mathbb{Q}^n : Each rational point $q \in \mathbb{Q}^n$ is encoded as a string $x \in \{0,1\}^*$ in a natural way. We then write $K(q)$ for $K(x)$. In this manner, $K(S)$, $\mathbf{m}(S)$, K -minimizers, and d -approximate K -minimizers are all defined for sets $S \subseteq \mathbb{Q}^n$.
2. Lifting to \mathbb{R}^n . For $S \subseteq \mathbb{R}^n$, we define $K(S) = K(S \cap \mathbb{Q}^n)$ and $\mathbf{m}(S) = \mathbf{m}(S \cap \mathbb{Q}^n)$. Similarly, a K -minimizer for S is a K -minimizer for $S \cap \mathbb{Q}^n$, etc.

For each $r \in \mathbb{N}$ and each $m = (m_1, \dots, m_n) \in \mathbb{Z}^n$, let

$$Q_m^{(r)} = [m_1 \cdot 2^{-r}, (m_1 + 1) \cdot 2^{-r}) \times \cdots \times [m_n \cdot 2^{-r}, (m_n + 1) \cdot 2^{-r})$$

be the r -dyadic cube at m . Note that each $Q_m^{(r)}$ is “half-open, half-closed” in such a way that, for each $r \in \mathbb{N}$, the family

$$\mathcal{Q}^{(r)} = \{Q_m^{(r)} \mid m \in \mathbb{Z}^n\}$$

is a partition of \mathbb{R}^n . It follows that (modulo trivial encoding) the collection

$$\mathcal{Q} = \{Q_m^{(r)} \mid r \in \mathbb{N} \text{ and } m \in \mathbb{Z}^n\}$$

of all *dyadic cubes* is a layered disjoint system whose r th layer is $\mathcal{Q}^{(r)}$. Moreover, the set

$$\{(r, m, q) \in \mathbb{N} \times \mathbb{Z}^n \times \mathbb{Q}^n \mid q \in Q_m^{(r)}\}$$

is decidable, so Theorem 2.3.1 has the following useful consequence.

Corollary 2.4.1. *There is a constant $c \in \mathbb{N}$ such that, for all $r, d \in \mathbb{N}$, no r -dyadic cube has more than $2^{d+K(r)+c}$ d -approximate K -minimizers. In particular, no r -dyadic cube has more than $2^{K(r)+c}$ K -minimizers.*

The Kolmogorov complexity of an arbitrary point in Euclidean space depends on both the point and a precision parameter.

Definition. Let $x \in \mathbb{R}^n$ and $r \in \mathbb{N}$. The *Kolmogorov complexity* of x at *precision* r is

$$K_r(x) = K(B_{2^{-r}}(x)).$$

That is, $K_r(x)$ is the number of bits required to specify *some* rational point in the open ball $B_{2^{-r}}(x)$. Note that, for each $q \in \mathbb{Q}^n$, $K_r(q) \nearrow K(q)$ as $r \rightarrow \infty$.

Given an open ball B of radius ρ and a real number $\alpha > 0$, we write αB for the ball with the same center as B and radius $\alpha\rho$. We also write \overline{B} for the topological closure of B .

The definition of $K_r(x)$ directs our attention to the Kolmogorov complexities of arbitrary balls of radius 2^{-r} in Euclidean space. The following easy fact is repeatedly useful in this context.

Observation 2.4.2. *For every open ball $B \subseteq \mathbb{R}^n$ of radius 2^{-r} ,*

$$B \cap 2^{-(r+\lceil \frac{1}{2} \log n \rceil)} \mathbb{Z}^n \neq \emptyset.$$

Proof. If B is such a ball, then the expanded ball

$$B' = 2^{r + \lceil \frac{1}{2} \log n \rceil} B$$

has radius

$$2^{\lceil \frac{1}{2} \log n \rceil} > 2^{\frac{1}{2} \log n - 1} = \frac{\sqrt{n}}{2}.$$

This implies that

$$B' \cap \mathbb{Z}^n \neq \emptyset,$$

whence

$$\begin{aligned} B \cap 2^{-(r + \lceil \frac{1}{2} \log n \rceil)} \mathbb{Z}^n &= 2^{-(r + \lceil \frac{1}{2} \log n \rceil)} (B' \cap \mathbb{Z}^n) \\ &\neq \emptyset. \end{aligned} \quad \square$$

We use Observation 2.4.2 to establish the following connection between the complexities of cubes and the complexities of balls.

Lemma 2.4.3. *There is a constant $c \in \mathbb{N}$ such that, for every $r \in \mathbb{N}$, every r -dyadic cube Q , and every open ball $B \subseteq \mathbb{R}^n$ of radius 2^{-r} that intersects Q ,*

$$K(B) \leq K(Q) + K(r) + c.$$

Proof. Fix a computable enumeration m_0, m_1, m_2, \dots of \mathbb{Z}^n satisfying $|m_i| \leq |m_{i+1}|$ for all $i \in \mathbb{N}$. Note that, for all $i \in \mathbb{N}$,

$$i < |\overline{B_{|m_i|}(0)} \cap \mathbb{Z}^n| \leq (2|m_i| + 1)^n. \quad (2.4.1)$$

Let $l = \lceil \frac{1}{2} \log n \rceil$, and let M be a self-delimiting Turing machine such that, if $U(\pi_1) = q \in \mathbb{Q}^n$ and $U(\pi_2) = r \in \mathbb{N}$, then, for all $i \in \mathbb{N}$,

$$M(\pi_1 \pi_2 0^{|s_i|} 1 s_i) = q + 2^{-(r+l)} m_i. \quad (2.4.2)$$

Let $c = 2 \lceil 2n \log(1 + \sqrt{n}) \rceil + 1 + c_M$, where c_M is an optimality constant for M .

Now assume the hypothesis, and let q be a K -minimizer of Q . Observation 2.4.2 tells us that there is a point $m \in \mathbb{Z}^n$ such that $2^{-(r+l)}m \in B - q$. Then $|2^{-(r+l)}m|$ is the distance from a point in B to the point $q \in Q$, so

$$|m| = 2^{r+l}|2^{-(r+l)}m| \leq 2^{r+l} \text{diam}(B \cup Q).$$

Since $B \cap Q \neq \emptyset$, it follows that

$$\begin{aligned} |m| &\leq 2^{r+l}[\text{diam}(B) + \text{diam}(Q)] \\ &= 2^l(2 + \sqrt{n}) \\ &\leq \frac{\sqrt{n}}{2}(2 + \sqrt{n}) \\ &= \frac{n}{2} + \sqrt{n}. \end{aligned} \tag{2.4.3}$$

It is crucial here that this bound does not depend on B , Q , or r .

Choose $i \in \mathbb{N}$ such that $m_i = m$. By (2.4.1) and (2.4.3),

$$i < (2(\frac{n}{2} + \sqrt{n}) + 1)^n = (1 + \sqrt{n})^{2n}. \tag{2.4.4}$$

Now let $\pi = \pi_1\pi_20^{|s_i|}1s_i$, where π_1 and π_2 are minimum-length programs for q and r , respectively. By (2.4.2) we have

$$M(\pi) = q + 2^{-(r+l)}m_i \in B.$$

It follows by (2.4.4) that

$$\begin{aligned} K(B) &\leq K(q + 2^{-(r+l)}m_i) \\ &\leq K_M(q + 2^{-(r+l)}m_i) + c_M \\ &\leq |\pi| + c_M \\ &= K(q) + K(r) + 2|s_i| + 1 + c_M \\ &= K(Q) + K(r) + 2\lceil 2n \log(1 + \sqrt{n}) \rceil + 1 + c_M \\ &= K(Q) + K(r) + c. \end{aligned} \quad \square$$

Theorem 2.4.4. *There is a constant $c \in \mathbb{N}$ such that, for all $r, d \in \mathbb{N}$, no open ball of radius 2^{-r} has more than $2^{d+2K(r)+c}$ d -approximate K -minimizers. In particular, no open ball of radius 2^{-r} has more than $2^{2K(r)+c}$ K -minimizers.*

Proof. Let B be an open ball of radius 2^{-r} , let Q be a r -dyadic cube such that $B \cap Q = \emptyset$, and let $u = K(B) - K(Q)$. There are at most $2^{d+u+K(r)+c'}$ $(d+u)$ -approximate K -minimizers $q \in \mathbb{Q}$ of Q such that $K(q) \leq K(Q) + d + u = K(B) + d$ where $c' \in \mathbb{N}$ is a constant from Corollary 2.4.1. Therefore, there are at most $2^{d+u+K(r)+c'}$ d -approximate K -minimizers of B in $Q \cap B$.

Observe that it takes at most $3^n = 2^{n \log 3}$ r -dyadic cubes to cover B . By Lemma 2.4.3, $u \leq K(r) + c''$, where $c'' \in \mathbb{N}$ is a constant. Therefore, it follows that B has at most $2^{d+2K(r)+c}$ d -approximate K -minimizers where $c = c' + c'' + n \log 3$. In particular, B has at most $2^{2K(r)+c}$ K -minimizers. \square

2.5 Upper Bounds on $K_r(x)$ and $K_{r+s}(x)$

Lemma 2.4.3 gives a slightly simplified proof of the known upper bound on $K_r(x)$.

Observation 2.5.1 ([47]). *For all $x \in \mathbb{R}^n$, $K_r(x) \leq nr + o(r)$.*

Proof. Let c be a constant of Lemma 2.4.3, let $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, and let

$$\gamma_x = \max\{|x_i| + 1 \mid 1 \leq i \leq n\}.$$

For each $r \in \mathbb{N}$, let $m(r) = (m_1, \dots, m_n)$ be the unique $m \in \mathbb{Z}^n$ such that $x \in \mathbb{Q}_m^{(r)}$. Then, for each $r \in \mathbb{N}$ and $1 \leq i \leq n$, we have $|m_i| \leq 2^r \gamma_x$. It follows easily from this that there is a constant $c' \in \mathbb{N}$ such that, for every $r \in \mathbb{N}$,

$$K(m(r)) \leq n(\log(2^r \gamma_x) + 2 \log \log(2^r \gamma_x)) + c_1. \quad (2.5.1)$$

There is clearly a constant $c_2 \in \mathbb{N}$ such that, for every $r \in \mathbb{N}$,

$$K(2^{-r}m(r)) \leq K(m(r)) + K(r) + c_2. \quad (2.5.2)$$

By (2.5.1), (2.5.2), and Lemma 2.4.3 we now have

$$\begin{aligned}
K_r(x) &= K(B_{2^{-r}}(x)) \\
&\leq K(Q_{m(r)}^{(r)}) + K(r) + c \\
&\leq K(m(r)) + K(r) + c \\
&\leq nr + \epsilon(r),
\end{aligned}$$

where

$$\begin{aligned}
\epsilon(r) &= n(\log \gamma_x + 2 \log \log(2^r \gamma_x)) + 2K(r) + c + c_1 + c_2. \\
&= o(r)
\end{aligned}$$

as $r \rightarrow \infty$. □

Lemma 2.5.2. *There is a constant $c \in \mathbb{N}$ such that, for all $r, s \in \mathbb{N}$, $x \in \mathbb{R}^n$, and $q \in B_{2^{-r}}(x)$,*

$$K_{r+s}(x) \leq K(q) + ns + K(r) + a_s,$$

where $a_s = K(s) + 2 \log(\lceil \frac{1}{2} \log n \rceil + s + 3) + n(\lceil \frac{1}{2} \log n \rceil + 3) + K(n) + 2 \log n + c$.

Proof. Fix a computable enumeration m_0, m_1, m_2, \dots of \mathbb{Z}^n satisfying $|m_i| \leq |m_{i+1}|$ for all $i \in \mathbb{N}$. Note that, for all $i \in \mathbb{N}$,

$$i < |\overline{B_{|m_i|}(0)} \cap \mathbb{Z}^n| \leq (2|m_i| + 1)^n. \quad (2.5.3)$$

Let $l = \lceil \frac{1}{2} \log n \rceil$, and let M be a self-delimiting Turing machine such that, if $U(\pi_1) = q \in \mathbb{Q}^n$, $U(\pi_2) = r \in \mathbb{N}$, $U(\pi_3) = s \in \mathbb{N}$, $U(\pi_4) = n \in \mathbb{N}$, and $U(\pi_5) = i \in \mathbb{N}$, then

$$M(\pi_1 \pi_2 \pi_3 \pi_4 \pi_5) = q + 2^{-(r+s+l+1)} m_i. \quad (2.5.4)$$

Let $a_s = 2n(\lceil \frac{1}{2} \log n \rceil + s + 3) + 1 + c_M$, where c_M is an optimality constant for M .

Now assume the hypothesis. Observation 2.4.2 tells us that there is a point $m \in \mathbb{Z}^n$ such that $2^{-(r+s+l)}m \in B_{2^{-(r+s)}}(x) - q$. Then $|2^{-(r+s+l)}m|$ is the distance from a point in $B_{2^{-(r+s)}}(x)$ to the point q , so

$$\begin{aligned}
|m| &= 2^{r+s+l}|2^{-(r+s+l)}m| \\
&\leq 2^{r+s+l}(2^{-r} + 2^{-(r+s)}) \\
&= 2^{s+l}(1 + 2^{-s}) \\
&= 2^l(2^s + 1) \\
&\leq 2^l 2^{s+1} \\
&\leq 2^{l+s+1}.
\end{aligned} \tag{2.5.5}$$

Choose $i \in \mathbb{N}$ such that $m_i = m$. By (2.5.3) and (2.5.5),

$$i < (2|m_i| + 1)^n \leq (2(2^{l+s+1}) + 1)^n = (2^{l+s+2} + 1)^n. \tag{2.5.6}$$

Now let $\pi = \pi_1\pi_2\pi_3\pi_4\pi_5$, where $\pi_1, \pi_2, \pi_3, \pi_4$, and π_5 are minimum-length programs for q, r, s, n , and i , respectively. By (2.5.4) we have

$$M(\pi) = q + 2^{-(r+s+l+1)}m_i \in B_{2^{-(r+s)}}(x). \tag{2.5.7}$$

Therefore, (2.5.7) and optimality tell us that

$$\begin{aligned}
K_{r+s}(x) &= K(B_{2^{-(r+s)}}(x)) \\
&\leq K(q + 2^{-(r+l)}m_i) \\
&\leq K_M(q + 2^{-(r+l)}m_i) + c_M \\
&= |\pi| + c_M \\
&= K(q) + K(r) + K(s) + K(n) + K(i) + c_M.
\end{aligned}$$

As noted in section 2, there is a constant $c_0 \in \mathbb{N}$ such that

$$K(i) \leq \log(1 + i) + 2\log(1 + \log(1 + i)) + c_0.$$

It follows by (2.5.6) that

$$\begin{aligned}
K(i) &\leq n \log(1 + 2^{l+s+2}) + 2 \log(1 + n \log(1 + 2^{l+s+2})) + c_0 \\
&\leq n(l + s + 3) + 2 \log(1 + n(l + s + 3)) + c_0 \\
&\leq n(l + s + 3) + 2(1 + \log n + \log(l + s + 3)) + c_0 \\
&= ns + n(l + 3) + 2 \log n + 2 \log(l + s + 3) + c_0 + 2.
\end{aligned}$$

Letting $c = c_M + c_0 + 2$, it follows that

$$K_{r+s}(x) \leq K(q) + ns + a_s,$$

where $a_s = K(s) + 2 \log(l + s + 3) + n(l + 3) + K(n) + 2 \log n + c$. \square

The following corollary says roughly that, in \mathbb{R}^n , precision can be improved by ns bits by adding ns bits of specification.

Corollary 2.5.3. *There is a constant $c \in \mathbb{N}$ such that, for all $r, s \in \mathbb{N}$ and $x \in \mathbb{R}^n$,*

$$K_{r+s}(x) \leq K_r(x) + ns + b_s,$$

where $b_s = a_s + K(r)$ and a_s is as in Lemma 2.5.2.

2.6 Algorithmic Mutual Information in Euclidean Space

This section develops the algorithmic mutual information between points in Euclidean space at a given precision. As previously discussed, we assume that rational points $q \in \mathbb{Q}^n$ are encoded as binary strings in some natural way. Mutual information between rational points is then defined from conditional Kolmogorov complexity in the standard way [42] as follows.

Definition. Let $p \in \mathbb{Q}^m$, $r \in \mathbb{Q}^n$, $s \in \mathbb{Q}^t$.

1. The *mutual information* between p and q is

$$I(p : q) = K(q) - K(q | p).$$

2. The *mutual information* between p and q given s is

$$I(p : q | s) = K(q | s) - K(q | p, s).$$

The following properties of mutual information are well known [42].

Theorem 2.6.1. Let $p \in \mathbb{Q}^m$ and $q \in \mathbb{Q}^n$.

1. $I(p, K(p) : q) = K(p) + K(q) - K(p, q) + O(1)$.
2. $I(p, K(p) : q) = I(q, K(q) : p) + O(1)$.
3. $I(p : q) \leq \min \{K(p), K(q)\} + O(1)$.

(Each of the properties 1 and 2 above is sometimes called *symmetry of mutual information*.)

Mutual information between points in Euclidean space at a given precision is now defined as follows.

Definition. The *mutual information* of $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ at *precision* $r \in \mathbb{N}$ is

$$I_r(x : y) = \min \{I(q_x : q_y) \mid q_x \in B_{2^{-r}}(x) \cap \mathbb{Q}^n \text{ and } q_y \in B_{2^{-r}}(y) \cap \mathbb{Q}^t\}.$$

As noted in the introduction, the role of the minimum in the above definition is to eliminate “spurious” information that points $q_x \in B_{2^{-r}} \cap \mathbb{Q}^n$ and $q_y \in B_{2^{-r}}(y) \cap \mathbb{Q}^t$ might share for reasons not forced by their proximities to x and y , respectively.

Notation. We also use the quantity

$$\begin{aligned} & J_r(x : y) \\ &= \min \{I(p_x : p_y) \mid p_x \text{ is a K-minimizer of } B_{2^{-r}}(x) \text{ and } p_y \text{ is a K-minimizer of } B_{2^{-r}}(y)\}. \end{aligned}$$

2.7 Relating $I_r(x : y)$ and $J_r(x : y)$

Although $J_r(x : y)$, having two “layers of minimization”, is somewhat more involved than $I_r(x : y)$, one can imagine using it as the definition of mutual information. In fact, for all $x, y \in \mathbb{R}$, $J_r(x : y)$ does not differ greatly from $I_r(x : y)$. We next develop machinery for proving this useful fact, which is Theorem 2.7.7 below.

Lemma 2.7.1. *There is a constant $c \in \mathbb{N}$ such that, for any $r \in \mathbb{N}$, open ball $B \subseteq \mathbb{R}^n$ of radius 2^{-r} , and $q \in B \cap \mathbb{Q}^n$,*

$$|\{p' \in B_{2^{1-r}}(q) \cap \mathbb{Q}^n \mid K(p') \leq K(B)\}| \leq 2^{K(r)+2K(r-1)+c}.$$

Proof. Let B be centered at $x \in \mathbb{R}^n$. If $p_q \in \mathbb{Q}^n$ is a K -minimizer of $B_{2^{1-r}}(q)$, then $p_q \in B_{2^{2-r}}(x)$. By Lemma 2.5.2,

$$\begin{aligned} K(B) &\leq K(p_q) + K(r) + c \\ &= K(B_{2^{1-r}}(q)) + K(r) + c, \end{aligned}$$

where $c = K(2) + K(n) + 2n(\lceil \frac{1}{2} \log n \rceil + 5) + 1 + c'$ for some constant c' . This inequality implies that any K -minimizer of B is also a $K(r) + c$ -approximate K -minimizer of $B_{2^{1-r}}(q)$. Therefore, by Lemma 2.4.4,

$$\begin{aligned} &|\{p' \in B_{2^{1-r}}(q) \cap \mathbb{Q}^n \mid K(p') \leq K(B)\}| \\ &\leq |\{p' \in B_{2^{1-r}}(q) \cap \mathbb{Q}^n \mid K(p') \leq K(B_{2^{1-r}}(q)) + K(r) + c\}| \\ &\leq 2^{K(r)+2K(r-1)+c}. \end{aligned} \quad \square$$

Lemma 2.7.2. *For all $x \in \mathbb{R}^n$, $q \in \mathbb{Q}^t$, and $q_x, p_x \in B_{2^{-r}}(x) \cap \mathbb{Q}^n$ where p_x is a K -minimizer of $B_{2^{-r}}(x)$,*

$$K(q \mid q_x) \leq K(q \mid p_x) + K(K(p_x)) + o(r).$$

Proof. Let M be a self-delimiting Turing machine that takes programs of the form $\pi = \langle \pi_1 \pi_2 \pi_3 0^{|s_i|} 1 s_i, q \rangle$, where $U(\pi_1, p) = q' \in \mathbb{Q}^t$, $U(\pi_2) = K(p)$, $U(\pi_3) = r \in \mathbb{N}$, and $i \in \mathbb{N}$.

M runs π_2 and π_3 on U to obtain $K(p)$ and r , performs a systematic search for the i^{th} discovered element of $\{p' \in B_{2^{1-r}}(q) \cap \mathbb{Q}^n \mid K(p') \leq K(p)\}$, and outputs $U(\langle \pi_1, p_i \rangle)$.

Therefore,

$$M(\pi) = U(\langle \pi_1, p_i \rangle). \quad (2.7.1)$$

Let c_M be an optimality constant for M .

Assume the hypothesis, and let $\pi = \langle \pi_1 \pi_2 \pi_3 0^{|s_i|} 1 s_i, q_x \rangle$, where π_1 is a minimum-length program for q when given p_x , π_2 is a minimum-length program for $K(p_x)$, π_3 is a minimum-length program for r , and i is an index for p_x in the set $\{p' \in B_{2^{1-r}}(q_x) \cap \mathbb{Q}^n \mid K(p') \leq K(p_x)\}$. By (2.7.1), we have $M(\pi) = U(\langle \pi_1, p_x \rangle) = q$. Therefore, by Lemma 2.7.1 and optimality,

$$\begin{aligned} K(q \mid q_x) &\leq K_M(q \mid q_x) + c_M \\ &\leq |\pi_1 \pi_2 \pi_3 0^{|s_i|} 1 s_i| + c_M \\ &= K(q \mid p_x) + K(K(p_x)) + K(r) + 2|s_i| + 1 + c_M \\ &\leq K(q \mid p_x) + K(K(p_x)) + K(r) + 2 \log |\{p' \in B_{2^{1-r}}(q_x) \cap \mathbb{Q}^n \mid K(p') \leq K(p_x)\}| \\ &\quad + 1 + c_M \\ &\leq K(q \mid p_x) + K(K(p_x)) + K(r) + 2(K(r) + 2K(r-1) + c) + 1 + c_M \\ &= K(q \mid p_x) + K(K(p_x)) + o(r). \end{aligned} \quad \square$$

By Lemma 2.7.2 and Observation 2.5.1 we have the following.

Corollary 2.7.3. *Let $x \in \mathbb{R}^n$. If $q_x \in B_{2^{-r}}(x) \cap \mathbb{Q}^n$ and $p_x \in \mathbb{Q}^n$ is a K -minimizer of $B_{2^{-r}}(x)$, then $K(p_x \mid q_x) = o(r)$.*

Lemma 2.7.4. *Let $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$. If $p_x \in B_{2^{-r}}(x)$ and $q_y, p_y \in B_{2^{-r}}(y)$ where p_x is a K -minimizer for $B_{2^{-r}}(x)$ and p_y is a K -minimizer for $B_{2^{-r}}(y)$, then*

$$K(p_x \mid q_y, K(q_y)) \leq K(p_x \mid p_y, K(p_y)) + o(r).$$

Proof. By the triangle inequality for strings and Corollary 2.7.3,

$$\begin{aligned} K(p_x | q_y, K(q_y)) &\leq K(p_x | p_y, K(p_y)) + K(p_y | q_y, K(q_y)) + O(1) \\ &\leq K(p_x | p_y, K(p_y)) + K(p_y | q_y) + O(1) \\ &= K(p_x | p_y, K(p_y)) + o(r). \end{aligned} \quad \square$$

The following lemma was inspired by Hammer et al. [26].

Lemma 2.7.5. *For all $x, y, z \in \{0, 1\}^*$,*

$$\begin{aligned} K(z) - K(K(z)) - K(K(x)) &\leq I(x : y) + K(z | x, K(x)) + K(z | y, K(y)) \\ &\quad - K(z | \langle x, y \rangle, K(\langle x, y \rangle)) - I(x : y | z) + O(1). \end{aligned}$$

Proof. By the well-known identity (2.1.1), obvious inequalities, and basic definitions.

$$\begin{aligned} &K(z) - K(K(z)) - K(K(x)) \\ &= K(x) - K(x, y) - K(K(x)) + K(x, z) - K(x) + K(y, z) - K(x, y, z) \\ &\quad + K(x, y) + K(z) - K(z, y) - K(K(z)) + K(x, z, y) - K(x, z) + O(1) \\ &= -K(y | x, K(x)) - K(K(x)) + K(x, z) - K(x) + K(y, z) - K(x, y, z) \\ &\quad + K(x, y) - K(y | z, K(z)) - K(K(z)) + K(y | x, z, K(x, z)) + O(1) \\ &\leq K(y) - K(y | x) + K(x, z) - K(x) + K(y, z) - K(y) - K(x, y, z) + K(x, y) \\ &\quad - K(y | z) + K(y | x, z) + O(1) \\ &= I(x : y) + K(z | x, K(x)) + K(z | y, K(y)) - K(z | x, y, K(x, y)) - I(x : y | z) + O(1). \end{aligned}$$

□

Corollary 2.7.6. *For all $x, y, z \in \{0, 1\}^*$,*

$$I(x : y) \geq K(z) - K(z | x, K(x)) - K(z | y, K(y)) - K(K(x)) - K(K(z)) + O(1).$$

Theorem 2.7.7. *For all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$I_r(x : y) = J_r(x : y) + o(r).$$

Proof. Let $q_x, p_x \in \mathbb{Q}^n$ and $q_y, p_y \in \mathbb{Q}^t$ where p_x is a K -minimizer of $B_{2^{-r}}(x)$, p_y is a K -minimizer of $B_{2^{-r}}(y)$, and $I(q_x : q_y) = I_r(x : y)$. By Lemma 2.7.2,

$$K(q_y) - K(q_y | p_x) \leq K(q_y) - K(q_y | q_x) + K(K(p_x)) + o(r).$$

Applying the definition of mutual information for rationals, we have

$$I(p_x : q_y) \leq I(q_x : q_y) + K(K(p_x)) + o(r),$$

which, by Corollary 2.7.6 and Observation 2.5.1, implies that

$$\begin{aligned} I(q_x : q_y) &\geq K(p_x) - K(p_x | p_x, K(p_x)) - K(p_x | q_y, K(q_y)) + o(r) \\ &= K(p_x) - K(p_x | q_y, K(q_y)) + o(r). \end{aligned}$$

By applying Lemma 2.7.4 and the definition of mutual information for rationals to the above inequality, we obtain

$$\begin{aligned} I(q_x : q_y) &\geq K(p_x) - K(p_x | p_y, K(p_y)) + o(r) \\ &= I(p_y, K(p_y) : p_x) + o(r). \end{aligned}$$

Thus, by Theorem 2.6.1,

$$\begin{aligned} I(q_x : q_y) &\geq I(p_x, K(p_x) : p_y) + o(r) \\ &\geq I(p_x : p_y) + o(r). \end{aligned}$$

The above inequality tells us that $I_r(x : y) = I(q_x : q_y) \geq I(p_x : p_y) + o(r) = J_r(x : y) + o(r)$. Also, by definition, $I_r(x : y) \leq J_r(x : y)$. \square

Before discussing the properties of $I_r(x : y)$, we need one more lemma.

Lemma 2.7.8. *Let $x \in \mathbb{R}^n$, $y \in \mathbb{R}^t$, and $r \in \mathbb{N}$. If $p_x \in \mathbb{Q}^n$ is a K -minimizer of $B_{2^{-r}}(x)$ and $p_y \in \mathbb{Q}^t$ is a K -minimizer of $B_{2^{-r}}(y)$, then*

$$K(p_x, p_y) = K_r(x, y) + o(r).$$

Proof. By Corollary 2.7.3,

$$\begin{aligned}
 K_r(x, y) &\leq K(p_x, p_y) \leq K(p_y) + K(p_x | p_y) \\
 &= K_r(y) + K(p_x | p_y) \\
 &\leq K_r(x, y) + K(p_x | p_y) + O(1) \\
 &= K_r(x, y) + o(r). \quad \square
 \end{aligned}$$

2.8 Properties of $I_r(x : y)$

The following characterization of algorithmic (Martin-Löf) randomness is well known.

Definition. A point $x \in \mathbb{R}^n$ is *random* if there is a constant $d \in \mathbb{N}$ such that, for all $r \in \mathbb{N}$,

$$K_r(x) \geq nr - d.$$

Two points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ are *independently random* if the point $(x, y) \in \mathbb{R}^{n+t}$ is random.

We now establish the following useful properties of $I_r(x : y)$.

Theorem 2.8.1. For all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,

1. $I_r(x : y) = K_r(x) + K_r(y) - K_r(x, y) + o(r)$.
2. $I_r(x : y) \leq \min\{K_r(x), K_r(y)\} + o(r)$.
3. If x and y are independently random, then $I_r(x : y) = o(r)$.
4. $I_r(x : y) = I_r(y : x) + o(r)$.

Proof. To prove the first statement, let $p_x \in \mathbb{Q}^n$ be a K -minimizer of $B_{2-r}(x)$ and $p_y \in \mathbb{Q}^t$ be a K -minimizer of $B_{2-r}(y)$. First, by Theorem 2.7.7,

$$\begin{aligned} I_r(x : y) &= J_r(x : y) + o(r) \\ &= I(p_x : p_y) + o(r) \\ &= K(p_y) - K(p_y | p_x) + o(r) \\ &\leq K(p_y) - K(p_y | p_x, K(p_x)) + o(r). \end{aligned}$$

By (2.1.1) and Lemma 2.7.8, this implies that

$$\begin{aligned} I_r(x : y) &\leq K(p_y) + K(p_x) - K(p_x, p_y) + o(r) \\ &= K_r(y) + K_r(x) - K_r(x, y) + o(r). \end{aligned}$$

Next we show that $I_r(x : y) \geq K_r(x) + K_r(y) - K_r(x, y) + o(r)$. By the above inequality,

$$\begin{aligned} I_r(x : y) &= K(p_y) - K(p_y | p_x) + o(r) \\ &\geq K(p_y) - K(p_y | p_x, K(p_x)) - K(K(p_x)) + o(r). \end{aligned}$$

Finally, by (2.1.1), Observation 2.5.1, and Lemma 2.7.8,

$$\begin{aligned} I_r(x : y) &\geq K(p_y) + K(p_x) - K(p_x, p_y) + o(r) \\ &\geq K_r(y) + K_r(x) - K_r(x, y) + o(r). \end{aligned}$$

We continue to the second statement. By 1,

$$\begin{aligned} I_r(x : y) &= K_r(x) + K_r(y) - K_r(x, y) + o(r) \\ &\leq K_r(x) + K_r(y) - K_r(y) + o(r) \\ &= K_r(x) + o(r). \end{aligned}$$

Likewise, $I_r(x : y) \leq K_r(y) + o(r)$. Therefore, $I_r(x : y) \leq \min\{K_r(x), K_r(y)\} + o(r)$.

We now prove the third statement. By 1,

$$\begin{aligned}
I_r(x : y) &= K_r(x) + K_r(y) - K_r(x, y) + o(r) \\
&\leq K_r(x) + K_r(y) + K(r) - K_r(r, x, y) + o(r) \\
&\leq nr + tr + K(r) - (n + t)r + o(r) \\
&= o(r),
\end{aligned}$$

where the last inequality is due to the premise that x and y are independently random and Observation 2.5.1.

Lastly, we prove the fourth statement. By 1 and Lemma 2.7.8,

$$\begin{aligned}
I_r(x : y) &= K_r(x) + K_r(y) - K_r(x, y) + o(r) \\
&= K_r(x) + K_r(y) - K(p_x, p_y) + o(r) \\
&= K_r(x) + K_r(y) - K(p_y, p_x) + o(r) \\
&= K_r(x) + K_r(y) - K_r(y, x) + o(r) \\
&= I_r(y : x) + o(r).
\end{aligned}$$

□

CHAPTER 3. MUTUAL DIMENSION AND DATA PROCESSING INEQUALITIES

In the previous chapter, we defined the mutual information between two points in Euclidean space at a given precision and explored its basic properties. In this chapter, we define the *lower* and *upper mutual dimensions* between two points in Euclidean space. Intuitively, this is the density of algorithmic mutual information between two points. In Section 3.1, we show that mutual dimension has *all* of the properties one would expect a measure of mutual information to have [3] with the exception of a data processing inequality.

Section 3.3 is dedicated to exploring various data processing inequalities for mutual dimension in Euclidean space. Roughly speaking, a data processing inequality states that the amount of shared information between two objects cannot be significantly increased when one of the objects is processed by a particular class of functions. We show that, if a computable function $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is Lipschitz continuous, then, for all $x, y \in \mathbb{R}^n$, the mutual dimension between $f(x)$ and y is *no greater than* the mutual dimension between x and y . We also demonstrate how to obtain other data processing inequalities by placing different continuity restrictions on f .

Section 3.4 investigates *reverse* data processing inequalities in Euclidean space, i.e., data processing inequalities where the function may significantly increase the mutual dimension between two points.

This chapter is a joint work with Jack H. Lutz and appeared in [10].

3.1 Mutual Dimension in Euclidean Space

We now define the lower and upper mutual dimensions between points in Euclidean space(s).

Definition. The *lower* and *upper mutual dimensions* between $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$ are

$$mdim(x : y) = \liminf_{r \rightarrow \infty} \frac{I_r(x : y)}{r}$$

and

$$Mdim(x : y) = \limsup_{r \rightarrow \infty} \frac{I_r(x : y)}{r},$$

respectively.

With the exception of the data processing inequality, which we prove in section 3.3, the following theorem says that the mutual dimensions $mdim$ and $Mdim$ have the basic properties that any mutual information measure should have. (See, for example, [3].)

Theorem 3.1.1. *For all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$, the following hold.*

1. $dim(x) + dim(y) - Dim(x, y) \leq mdim(x : y) \leq Dim(x) + Dim(y) - Dim(x, y)$.
2. $dim(x) + dim(y) - dim(x, y) \leq Mdim(x : y) \leq Dim(x) + Dim(y) - dim(x, y)$.
3. $mdim(x : y) \leq \min\{dim(x), dim(y)\}$, $Mdim(x : y) \leq \min\{Dim(x), Dim(y)\}$.
4. $0 \leq mdim(x : y) \leq Mdim(x : y) \leq \min\{n, t\}$.
5. *If x and y are independently random, then $Mdim(x : y) = 0$.*
6. $mdim(x : y) = mdim(y : x)$, $Mdim(x : y) = Mdim(y : x)$.

Proof. To prove the first statement, we use Theorem 2.8.1 and basic properties of \limsup and \liminf . First we show that $mdim(x : y) \geq dim(x) + dim(y) - Dim(x, y)$.

$$\begin{aligned}
mdim(x : y) &= \liminf_{r \rightarrow \infty} \frac{I_r(x : y)}{r} \\
&= \liminf_{r \rightarrow \infty} \frac{K_r(x) + K_r(y) - K_r(x, y) + o(r)}{r} \\
&\geq \liminf_{r \rightarrow \infty} \frac{K_r(x)}{r} + \liminf_{r \rightarrow \infty} \frac{K_r(y)}{r} + \liminf_{r \rightarrow \infty} \frac{-K_r(x, y)}{r} + \liminf_{r \rightarrow \infty} \frac{o(r)}{r} \\
&= dim(x) + dim(y) - \limsup_{r \rightarrow \infty} \frac{K_r(x, y)}{r} \\
&= dim(x) + dim(y) - Dim(x, y).
\end{aligned}$$

Next we show that $mdim(x : y) \leq Dim(x) + Dim(y) - Dim(x, y)$.

$$\begin{aligned}
mdim(x : y) &= Dim(x) + Dim(y) - Dim(x) - Dim(y) + mdim(x : y) \\
&= Dim(x) + Dim(y) \\
&\quad - \left(\limsup_{r \rightarrow \infty} \frac{K_r(x)}{r} + \limsup_{r \rightarrow \infty} \frac{K_r(y)}{r} + \limsup_{r \rightarrow \infty} \frac{-I_r(x : y)}{r} \right) \\
&\leq Dim(x) + Dim(y) \\
&\quad - \limsup_{r \rightarrow \infty} \frac{K_r(x) + K_r(y) - K_r(x) - K_r(y) + K_r(x, y) + o(r)}{r} \\
&= Dim(x) + Dim(y) - Dim(x, y).
\end{aligned}$$

The proof of the second statement is similar to the first. The third statement follows immediately from Theorem 2.8.1 and the fact that

$$\liminf_{r \rightarrow \infty} \min\{K_r(x), K_r(y)\} \leq \min\{\liminf_{r \rightarrow \infty} K_r(x), \liminf_{r \rightarrow \infty} K_r(y)\}.$$

The fourth statement follows from the third and the fact that, for all $x \in \mathbb{R}^n$, $Dim(x) \leq n$. Finally, both the fifth and sixth statements follow immediately from Theorem 2.8.1.

□

3.2 Computable Functions in Euclidean Space

In order to discuss data processing inequalities for points in Euclidean space, we must first understand what it means for a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ to be computable. In this chapter, we use well-known concepts from computable analysis found in [6, 35, 73].

An *oracle* for a point $x \in \mathbb{R}^n$ is a function $g_x : \mathbb{N} \rightarrow \mathbb{Q}^n$ such that,

$$|g_x(r) - x| \leq 2^{-r},$$

for all $r \in \mathbb{N}$. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is *computable* if there exists an oracle machine M such that, for every $x \in \mathbb{R}^n$ and every oracle g_x for x ,

$$|M^{g_x}(r) - f(x)| \leq 2^{-r},$$

for all $r \in \mathbb{N}$, i.e., M^{g_x} is an oracle for $f(x)$.

3.3 Data Processing Inequalities for Points in Euclidean Space

Our objectives in this section are to prove data processing inequalities for lower and upper mutual dimensions in Euclidean space.

Definition. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is *Lipschitz* if there is a constant $c > 0$ such that, for all $x, y \in \mathbb{R}^n$,

$$|f(x) - f(y)| \leq c|x - y|.$$

The following result is the main theorem of this chapter. The meaning and necessity of the Lipschitz hypothesis are explained in the introduction.

Theorem 3.3.1 (data processing inequality). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is computable and Lipschitz, then, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : y) \leq mdim(x : y)$$

and

$$Mdim(f(x) : y) \leq Mdim(x : y).$$

We in fact prove a stronger result.

Definition. A *modulus (of uniform continuity)* for a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is a nondecreasing function $m : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $x, y \in \mathbb{R}^n$ and $r \in \mathbb{N}$,

$$|x - y| \leq 2^{-m(r)} \Rightarrow |f(x) - f(y)| \leq 2^{-r}.$$

Note that it is well known that a function is uniformly continuous if and only if it has a modulus of uniform continuity.

Lemma 3.3.2. *If $f : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ is a computable function, then, for all $x, y, z \in \Sigma^*$,*

$$K(y|x) \leq K(y|f(x, z)) + K(z) + O(1).$$

Proof. Let M be a self-delimiting Turing machine such that if $U(\pi_1, f(x, z)) = y$, $U(\pi_2) = z$, and π_3 is a program for f where $x, y, z \in \Sigma^*$ and $f : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ is a partial computable function, then

$$M(\pi_1\pi_2\pi_3, x) = y. \quad (3.3.1)$$

Assume the hypothesis, and let $\pi = \pi_1\pi_2\pi_3$ where π_1 is a minimum-length program for y given $f(x, z)$, π_2 is a minimum-length program for z , and π_3 is a minimum-length program for f . Therefore, by (3.3.1), we have $M(\pi, x) = y$. By optimality,

$$\begin{aligned} K(y|x) &\leq K_M(y|x) + c_M \\ &\leq |\pi| + c_M \\ &= K(y|f(x, z)) + K(z) + K(f) + c_M \\ &= K(y|f(x, z)) + K(z) + O(1). \quad \square \end{aligned}$$

Lemma 3.3.3. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and $m : \mathbb{N} \rightarrow \mathbb{N}$ is a computable, strictly increasing modulus for f , then for every $x \in \mathbb{R}^n$, $y \in \mathbb{R}^k$,*

$$I_r(f(x) : y) \leq I_{m(r+1)}(x : y) + o(r).$$

Proof. Let $q_x \in \mathbb{Q}^n$ and $q_y \in \mathbb{Q}^t$ such that $I_{m(r+1)}(x : y) = I(q_x : q_y)$. Because $|x - q_x| \leq 2^{-m(r+1)}$, where m is a modulus for f , we know that $|f(x) - f(q_x)| \leq 2^{-(r+1)}$. Also, since f is computable, there exists an oracle Turing machine M that uses an oracle q_x such that $|M^{q_x}(r) - f(q_x)| \leq 2^{-r}$. Let $h : \mathbb{N} \times \mathbb{Q}^n \rightarrow \mathbb{Q}^k$ be a function such that $h(q_x, r) = M^{q_x}(r + 1)$. Observe that

$$\begin{aligned} |M^{q_x}(r + 1) - f(x)| &\leq |f(x) - f(q_x)| + |M^{q_x}(r + 1) - f(q_x)| \\ &\leq 2^{-(r+1)} + 2^{-(r+1)} \\ &= 2^{-r}. \end{aligned}$$

From this and Lemma 3.3.2, it follows that

$$\begin{aligned} I_r(f(x) : y) &\leq I(M^{q_x}(r + 1) : q_y) \\ &= I(h(q_x, r) : q_y) \\ &\leq I(q_x : q_y) + K(r) + O(1) \\ &= I_{m(r+1)}(x : y) + o(r). \end{aligned} \quad \square$$

Lemma 3.3.4 (modulus processing lemma). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and $m : \mathbb{N} \rightarrow \mathbb{N}$ is a computable, strictly increasing modulus for f , then for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : y) \leq mdim(x : y) \left(\limsup_{r \rightarrow \infty} \frac{m(r+1)}{r} \right)$$

and

$$Mdim(f(x) : y) \leq Mdim(x : y) \left(\limsup_{r \rightarrow \infty} \frac{m(r+1)}{r} \right),$$

except when $\left(\limsup_{r \rightarrow \infty} \frac{m(r+1)}{r} \right) = \infty$ while either $mdim(x : y) = 0$ or $Mdim(x : y) = 0$.

Proof. By Lemma 3.3.3, we have

$$\begin{aligned} mdim(f(x) : y) &\leq \liminf_{r \rightarrow \infty} \frac{I_{m(r+1)}(x : y)}{r} \\ &= \liminf_{r \rightarrow \infty} \left(\frac{I_{m(r+1)}(x : y)}{m(r+1)} \cdot \frac{m(r+1)}{r} \right) \\ &\leq mdim(x : y) \left(\limsup_{r \rightarrow \infty} \frac{m(r+1)}{r} \right). \end{aligned}$$

A similar proof can be given for $Mdim$. □

Theorem 3.3.1 follows immediately from Lemma 3.3.4 and the following well-known observation.

Observation 3.3.5. *A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is Lipschitz if and only if there exists $s \in \mathbb{N}$ such that $m(r) = r + s$ is a modulus for f .*

Definition. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is *Hölder with exponent $\alpha > 0$* if there is a constant $c > 0$ such that, for all $x, y \in \mathbb{R}^n$,

$$|f(x) - f(y)| \leq c|x - y|^\alpha.$$

We can derive an observation similar to Observation 3.3.5 for Hölder functions.

Observation 3.3.6. *If a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is Hölder with exponent α , then there exists $s \in \mathbb{N}$ such that $m(r) = \lceil \frac{1}{\alpha}(r + s) \rceil$ is a modulus for f .*

We can derive the following fact from Observation 3.3.6 and the modulus processing lemma.

Corollary 3.3.7. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and Hölder with exponent α , then, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : y) \leq \frac{1}{\alpha} mdim(x : y)$$

and

$$Mdim(f(x) : y) \leq \frac{1}{\alpha} Mdim(x : y).$$

3.4 Reverse Data Processing Inequalities

In this section we develop reverse versions of the data processing inequalities from section 3.3.

Notation. Let $n \in \mathbb{Z}^+$.

1. $[n] = \{1, \dots, n\}$.
2. For $S \subseteq [n]$, $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^{n-|S|}$, the string

$$x *_S y \in \mathbb{R}^n$$

is obtained by placing the components of x into the positions in S (in order) and the components of y into the positions in $[n] \setminus S$ (in order).

3. For each $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, let $x_{(i,j)} = (x_i, x_{i+1}, \dots, x_j)$ for every $i, j \in \mathbb{N}$ such that $i \leq j \leq n$.

Definition. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$.

1. f is *co-Lipschitz* if there is a real number $c > 0$ such that for all $x, y \in \mathbb{R}^n$,

$$|f(x) - f(y)| \geq c|x - y|.$$

2. f is *bi-Lipschitz* if f is both Lipschitz and co-Lipschitz.
3. For $S \subseteq [n]$, f is *S-co-Lipschitz* if there is a real number $c > 0$ such that, for all $u, v \in \mathbb{R}^{|S|}$ and $y \in \mathbb{R}^{n-|S|}$,

$$|f(u *_S y) - f(v *_S y)| \geq c|u - v|.$$

4. For $i \in [n]$, f is *co-Lipschitz in its i^{th} argument* if f is $\{i\}$ -co-Lipschitz.

Note that f is $[n]$ -co-Lipschitz if and only if f is co-Lipschitz.

Example. The function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ defined by

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n$$

is S -co-Lipschitz if and only if $|S| \leq 1$. In particular, if $n \geq 2$, then f is co-Lipschitz in every argument, but f is not co-Lipschitz.

We next relate co-Lipschitz conditions to moduli.

Definition. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$.

1. An *inverse modulus* for f is a nondecreasing function $m' : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $x, y \in \mathbb{R}^n$ and $r \in \mathbb{N}$,

$$|f(x) - f(y)| \leq 2^{-m'(r)} \Rightarrow |x - y| \leq 2^{-r}.$$

2. Let $S \subseteq [n]$. An *S -inverse modulus* for f is a nondecreasing function $m' : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $u, v \in \mathbb{R}^{|S|}$, all $y \in \mathbb{R}^{n-|S|}$, and all $r \in \mathbb{N}$,

$$|f(u *_S y) - f(v *_S y)| \leq 2^{-m'(r)} \Rightarrow |u - v| \leq 2^{-r}.$$

3. Let $i \in [n]$. An *inverse modulus for f in its i^{th} argument* is an $\{i\}$ -inverse modulus for f .

Observation 3.4.1. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $S \subseteq [n]$.

1. f is S -co-Lipschitz if and only if there is a positive constant $t \in \mathbb{N}$ such that $m'(r) = r + t$ is an S -inverse modulus of f .
2. f is co-Lipschitz if and only if there is a positive constant $t \in \mathbb{N}$ such that $m'(r) = r + t$ is an inverse modulus of f .

Definition. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ and $S \subseteq [n]$. We say that f is *S -injective* if, for all $x, y \in \mathbb{R}^n$ and $z \in \mathbb{R}^{n-|S|}$,

$$f(x *_S z) = f(y *_S z) \Rightarrow x = y.$$

Note f is injective if and only if f is $[n]$ -injective.

Definition. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ be a function and $S \subseteq [n]$ such that $n \in \mathbb{N}$. An S -left inverse of f is a partial function $g : \mathbb{R}^t \times \mathbb{R}^{n-|S|} \rightarrow \mathbb{R}^{|S|}$ such that, for all $x \in \mathbb{R}^{|S|}$ and $y \in \mathbb{R}^t \times \mathbb{R}^{n-|S|}$,

$$g(f(x *_S y), y) = x.$$

It is easy to prove that f has an S -left inverse if and only if f is S -injective.

Lemma 3.4.2. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ has an S -inverse modulus m' , then f is S -injective and m' is a modulus for any S -left inverse of f .*

Proof. Let $m' : \mathbb{N} \rightarrow \mathbb{N}$ be an S -inverse modulus for f , $x, y \in \mathbb{R}^{|S|}$ and $z \in \mathbb{R}^{n-|S|}$, then, if $f(x *_S z) = f(y *_S z)$,

$$|f(x *_S z) - f(y *_S z)| \leq 2^{-m'(r)},$$

for all $r \in \mathbb{N}$, which implies that

$$|x - y| \leq 2^{-r}.$$

Therefore, $x = y$ and f is S -injective.

Let $g : \mathbb{R}^t \times \mathbb{R}^{n-|S|} \rightarrow \mathbb{R}^{|S|}$ be an S -left inverse of f . Let $x, y \in \text{dom } g$ and $r \in \mathbb{N}$ such that $x = (f(u *_S w), w)$ and $y = (f(v *_S z), z)$, where $u, v \in \mathbb{R}^{|S|}$ and $w, z \in \mathbb{R}^{n-|S|}$. Assume that $|x - y| \leq 2^{-m'(r)}$, then

$$\begin{aligned} & |f(g(f(u *_S w), w) *_S w) - f(g(f(v *_S z), z) *_S z)| \\ &= |f(u *_S w) - f(v *_S z)| \\ &\leq |(f(u *_S w), w) - (f(v *_S z), z)| \\ &= |x - y| \\ &\leq 2^{-m'(r)}. \end{aligned}$$

So, $|g(f(u *_S w), w) - g(f(v *_S z), z)| \leq 2^{-r}$, and

$$\begin{aligned} |g(x) - g(y)| &= |g(f(u *_S w), w) - g(f(v *_S z), z)| \\ &\leq 2^{-r}. \end{aligned}$$

Therefore, m' is a modulus for g . □

Lemma 3.4.3. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^t$ is a computable and uniformly continuous function that has a computable S -inverse modulus m' , then f has a computable S -left inverse.*

Proof. Assume the hypothesis. Since f is computable and uniformly continuous, there exist a modulus m for f and an oracle Turing machine M_f such that, for every $x \in \mathbb{R}^n$, $r \in \mathbb{N}$, and every oracle h_x for x ,

$$|M_f^{h_x}(r) - f(x)| \leq 2^{-r}. \quad (3.4.1)$$

Define $g : \mathbb{R}^t \times \mathbb{R}^{n-|S|} \rightarrow \mathbb{R}^{|S|}$ by

$$g(z) = \begin{cases} x & \text{if } z = (f(x *_S y), y), \\ \text{undefined} & \text{if otherwise} \end{cases},$$

where $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^{n-|S|}$, and $z \in \mathbb{R}^t \times \mathbb{R}^{n-|S|}$.

We now show that g is computable. Let $z = (f(x *_S y), y) \in \text{dom } g$ and h_z be an oracle for z such that, for all $r \in \mathbb{N}$,

$$|h_z(r) - z| \leq 2^{-r}. \quad (3.4.2)$$

First we show that, for any $r \in \mathbb{N}$, there exist a rational $q \in \mathbb{Q}^{|S|}$ and an oracle h_{qy} for $q *_S y$ such that

$$|M_f^{h_{qy}}(m'(r) + 3) - h_z(m'(r) + 3)| \leq 2^{-(m'(r)+1)}.$$

Let $q \in \mathbb{Q}^{|S|}$ such that $|q *_S y - x *_S y| \leq 2^{-(m(m'(r)+2))}$, and let $h_{qy}(r) = q *_S h_z(r)_{(t+1, t+n-|S|)}$ be an oracle for $q *_S y$. Therefore,

$$|f(q *_S y) - f(x *_S y)| \leq 2^{-(m'(r)+2)}. \quad (3.4.3)$$

By (3.4.1), (3.4.2), (3.4.3),

$$\begin{aligned}
& |M_f^{h_{qy}}(m'(r) + 3) - h_z(m'(r) + 3)_{(1,t)}| \\
&= |M_f^{h_{qy}}(m'(r) + 3) - f(q *_S y) + f(q *_S y) - f(x *_S y) + f(x *_S y) - h_z(m'(r) + 3)_{(1,t)}| \\
&\leq |M_f^{h_{qy}}(m'(r) + 3) - f(q *_S y)| + |f(q *_S y) - f(x *_S y)| \\
&\quad + |h_z(m'(r) + 3)_{(1,t)} - f(x *_S y)| \\
&\leq 2^{-(m'(r)+3)} + 2^{-(m'(r)+2)} + 2^{-(m'(r)+3)} \\
&= 2^{-(m'(r)+1)}.
\end{aligned}$$

Let M_g be a Turing machine equipped with oracle h_z . Given an input $r \in \mathbb{N}$, M_g searches for and outputs a rational $q_x \in \mathbb{Q}^{|S|}$ such that

$$|M_f^{h_{qxy}}(m'(r) + 3) - h_z(m'(r) + 3)_{(1,t)}| \leq 2^{-(m'(r)+1)}, \quad (3.4.4)$$

where $h_{qxy} = q_x *_S h_z(r)_{(t+1, t+n-|S|)}$ is an oracle for $q_x *_S y$. We now show that $|M_g^{h_z}(r) - g(z)| \leq 2^{-r}$. By (3.4.1), (3.4.2), (3.4.4),

$$\begin{aligned}
& |f(M_g^{h_z}(r) *_S y) - f(x *_S y)| \\
&= |f(q_x *_S y) - f(x *_S y)| \\
&= |f(q_x *_S y) - M_f^{h_{qxy}}(m'(r) + 3) + M_f^{h_{qxy}}(m'(r) + 3) - h_z(m'(r) + 3)_{(1,t)} \\
&\quad + h_z(m'(r) + 3)_{(1,t)} - f(x *_S y)| \\
&\leq |f(q_x *_S y) - M_f^{h_{qxy}}(m'(r) + 3)| + |M_f^{h_{qxy}}(m'(r) + 3) - h_z(m'(r) + 3)_{(1,t)}| \\
&\quad + |h_z(m'(r) + 3)_{(1,t)} - f(x *_S y)| \\
&\leq 2^{-(m'(r)+3)} + 2^{-(m'(r)+1)} + 2^{-(m'(r)+3)} \\
&= 2^{-(m'(r)+2)} + 2^{-(m'(r)+1)} \\
&< 2^{-m'(r)}.
\end{aligned}$$

Since m' is an S -inverse modulus for f , we have

$$\begin{aligned} |M_g^{h_z}(r) - g(z)| &= |M_g^{h_z}(r) - x| \\ &\leq 2^{-r}. \end{aligned}$$

Therefore, g is a computable S -left inverse of f . \square

Lemma 3.4.4 (reverse modulus processing lemma). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is a computable and uniformly continuous function, and $m' : \mathbb{N} \rightarrow \mathbb{N}$ is a computable, strictly increasing S -inverse modulus for f , then, for all $S \subseteq [n]$, $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^t$, and $z \in \mathbb{R}^{n-|S|}$,*

$$mdim(x : y) \leq mdim((f(x *_S z), z) : y) \left(\limsup_{r \rightarrow \infty} \frac{m'(r+1)}{r} \right)$$

and

$$Mdim(x : y) \leq Mdim((f(x *_S z), z) : y) \left(\limsup_{r \rightarrow \infty} \frac{m'(r+1)}{r} \right),$$

except when $\left(\limsup_{r \rightarrow \infty} \frac{m'(r+1)}{r} \right) = \infty$ while either $mdim((f(x *_S z), z) : y) = 0$ or $Mdim((f(x *_S z), z) : y) = 0$.

Proof. Assume the hypothesis. By Lemmas 3.4.2 and 3.4.3, there exists a computable and uniformly continuous function g that is an S -left inverse of f and m' is a modulus for g . Then, for all $S \subseteq [n]$, $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^t$, and $z \in \mathbb{R}^{n-|S|}$,

$$mdim(x : y) = mdim(g(f(x *_S z), z) : y).$$

Therefore, by Lemma 3.3.4, we have

$$mdim(x : y) \leq mdim(f(x *_S z), z : y) \left(\limsup_{r \rightarrow \infty} \frac{m'(r+1)}{r} \right).$$

A similar proof can be given for $Mdim$. \square

By Observation 3.4.1 and Lemma 3.4.4, we have the following.

Theorem 3.4.5 (reverse data processing inequality). *If $S \subseteq [n]$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and S -co-Lipschitz, then, for all $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^t$, and $z \in \mathbb{R}^{n-|S|}$,*

$$mdim(x : y) \leq mdim((f(x *_S z), z) : y)$$

and

$$Mdim(x : y) \leq Mdim((f(x *_S z), z) : y).$$

Definition. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $0 < \alpha \leq 1$.

1. f is *co-Hölder with exponent α* if there is a real number $c > 0$ such that, for all $x, y \in \mathbb{R}^n$,

$$|x - y| \leq c|f(x) - f(y)|^\alpha.$$

2. For $S \subseteq [n]$, f is *S -co-Hölder with exponent α* if there is a real number $c > 0$ such that, for all $u, v \in \mathbb{R}^{|S|}$ and $y \in \mathbb{R}^{n-|S|}$,

$$|u - v| \leq c|f(u *_S y) - f(v *_S y)|^\alpha.$$

Observation 3.4.6. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $S \subseteq [n]$.*

1. *If f is S -co-Hölder with exponent α , then there exists $t \in \mathbb{N}$ such that $m'(r) = \lceil \frac{1}{\alpha}(r + t) \rceil$ is an S -inverse modulus of f .*
2. *If f is co-Hölder with exponent α , then there exists $t \in \mathbb{N}$ such that $m'(r) = \lceil \frac{1}{\alpha}(r + t) \rceil$ is an inverse modulus of f .*

The next corollary follows from the reverse modulus processing lemma and Observation 3.4.6.

Corollary 3.4.7. *If $S \subseteq [n]$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and S -co-Hölder with exponent α , then, for all $x \in \mathbb{R}^{|S|}$, $y \in \mathbb{R}^t$, and $z \in \mathbb{R}^{n-|S|}$,*

$$mdim(x : y) \leq \frac{1}{\alpha} mdim((f(x *_S z), z) : y)$$

and

$$Mdim(x : y) \leq \frac{1}{\alpha} Mdim((f(x *_S z), z) : y).$$

3.5 Data Processing Applications

In this section we use the data processing inequalities and their reverses to investigate how certain functions on Euclidean space preserve or predictably transform mutual dimensions.

Theorem 3.5.1 (mutual dimension conservation inequality). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $g : \mathbb{R}^t \rightarrow \mathbb{R}^l$ are computable and Lipschitz, then, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : g(y)) \leq mdim(x : y)$$

and

$$Mdim(f(x) : g(y)) \leq Mdim(x : y).$$

Proof. The conclusion follows from Theorem 3.1.1 and the data processing inequality.

$$\begin{aligned} mdim(f(x) : g(y)) &\leq mdim(x : g(y)) \\ &= mdim(g(y) : x) \\ &\leq mdim(y : x) \\ &= mdim(x : y). \end{aligned}$$

A similar argument can be given for $Mdim(f(x) : g(y)) \leq Mdim(x : y)$. □

Theorem 3.5.2 (reverse mutual dimension conservation inequality). *Let $S_1 \subseteq [n]$ and $S_2 \subseteq [t]$. If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and S_1 -co-Lipschitz, and $g : \mathbb{R}^t \rightarrow \mathbb{R}^l$ is computable and S_2 -co-Lipschitz, then, for all $x \in \mathbb{R}^{|S_1|}$, $y \in \mathbb{R}^{|S_2|}$, $w \in \mathbb{R}^{n-|S_1|}$, and $z \in \mathbb{R}^{t-|S_2|}$,*

$$mdim(x : y) \leq mdim((f(x *_S w), w) : (g(y *_S z), z))$$

and

$$Mdim(x : y) \leq Mdim((f(x *_S w), w) : (g(y *_S z), z)).$$

Proof. The conclusion follows from Theorem 3.1.1 and the reverse data processing inequality.

$$\begin{aligned}
mdim(x : y) &\leq mdim((f(x *_S w), w) : y) \\
&= mdim(y : (f(x *_S w), w)) \\
&\leq mdim((g(y *_S z), z) : (f(x *_S w), w)) \\
&= mdim((f(x *_S w), w) : (g(y *_S z), z)).
\end{aligned}$$

A similar argument can be given for $Mdim(x : y) \leq Mdim((f(x *_S w), w) : (g(y *_S z), z))$. \square

Corollary 3.5.3 (preservation of mutual dimension). *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $g : \mathbb{R}^t \rightarrow \mathbb{R}^l$ are computable and bi-Lipschitz, then, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : g(y)) = mdim(x : y)$$

and

$$Mdim(f(x) : g(y)) = Mdim(x : y).$$

Corollary 3.5.4. *If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $g : \mathbb{R}^t \rightarrow \mathbb{R}^l$ are computable and Hölder with exponents α and β , respectively, then, for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^t$,*

$$mdim(f(x) : g(y)) \leq \frac{1}{\alpha\beta} mdim(x : y)$$

and

$$Mdim(f(x) : g(y)) \leq \frac{1}{\alpha\beta} Mdim(x : y).$$

Corollary 3.5.5. *Let $S_1 \subseteq [n]$ and $S_2 \subseteq [t]$. If $f : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is computable and S_1 -co-Hölder with exponent α , and $g : \mathbb{R}^t \rightarrow \mathbb{R}^l$ is computable and S_2 -co-Hölder with exponent β , then, for all $x \in \mathbb{R}^{|S_1|}$, $y \in \mathbb{R}^{|S_2|}$, $w \in \mathbb{R}^{n-|S_1|}$, and $z \in \mathbb{R}^{t-|S_2|}$,*

$$mdim(x : y) \leq \frac{1}{\alpha\beta} mdim((f(x *_S w), w) : (g(y *_S z), z))$$

and

$$Mdim(x : y) \leq \frac{1}{\alpha\beta} Mdim((f(x *_S w), w) : (g(y *_S z), z)).$$

CHAPTER 4. BOUNDED TURING REDUCTIONS AND DATA PROCESSING INEQUALITIES FOR SEQUENCES

In Chapter 3, we defined and investigated mutual dimensions between points in Euclidean space. The purpose of this chapter is to develop a similar framework for the mutual dimension between sequences over an arbitrary alphabet.

The first half of this chapter defines the lower and upper mutual dimensions between sequences and shows that they are equal to the lower and upper mutual dimensions between the sequences' real representations, respectively. Using this result, we prove that mutual dimensions between sequences have nice properties.

The second half of this chapter addresses data processing inequalities for sequences. We show that for all sequences X, Y , and Z and all Turing functionals Φ such that $\Phi^X(n) = Z \upharpoonright n$, for all $n \in \mathbb{N}$, if the largest oracle query made during the computation of $\Phi^X(n)$ is bounded by $n + c$, where $c \in \mathbb{N}$ is a constant, then the mutual dimension between Z and Y is no greater than the mutual dimension between X and Y . We also derive other data processing inequalities by making adjustments to the computable bounds of the *use function* of Turing functionals.

We define the *yield* of a Turing functional Φ^S with access to at most n symbols of S to be the smallest $m \in \mathbb{N}$ such that $\Phi^{S \upharpoonright n}(m)$ does not halt and show how to derive reverse data processing inequalities by applying bounds to the Turing functional's yield.

Sections 4.1 through 4.6 are a joint work with Jack H. Lutz and can be found in [9]. The rest of this chapter can be found in [8].

4.1 Notation

Let $\Sigma = \{0, 1, \dots, k-1\}$ be the *alphabet* consisting of k symbols and Σ^* be the set of all strings over Σ . We write Σ^∞ for the set of all infinite sequences over Σ , and, for every $S \in \Sigma^\infty$ and $n \in \mathbb{N}$, $S[n]$ is the n th symbol of S and $S \upharpoonright n$ denotes the first n symbols of S . For all strings $x, y \in \Sigma^*$ and sequences $S \in \Sigma^\infty$, we write $x \sqsubseteq S$ and $x \sqsubseteq y$ to mean that x is a *prefix* of S and x is a prefix of y , respectively. For $S, T \in \Sigma^\infty$, the notation (S, T) represents the sequence in $(\Sigma \times \Sigma)^\infty$ obtained after pairing each symbol in S with the symbol in T located at the same position. For $S \in \Sigma^\infty$, let

$$\alpha_S = \sum_{i=0}^{\infty} S[i]k^{-(i+1)} \in [0, 1].$$

Informally, we say that α_S is the *real representation* of S .

4.2 Relating the Kolmogorov Complexities of Sequences to the Kolmogorov Complexities of Reals

In this section, we show that the Kolmogorov complexity of $S \upharpoonright r$ is equal to the Kolmogorov complexity of the real representation of S at precision r .

Recall the definition of the Kolmogorov complexity of a set of strings.

Definition (Shen and Vereshchagin [62]). The *Kolmogorov complexity* of a set $S \subseteq \Sigma^*$ is

$$K(S) = \min\{K(u) \mid u \in S\}.$$

Keeping in mind that tuples of rationals in \mathbb{Q}^n can be encoded as a string in Σ^* , we remind the reader of the definition of the Kolmogorov complexity of a real at precision r .

Definition. The *Kolmogorov complexity* of $x \in \mathbb{R}$ at precision $r \in \mathbb{N}$ is

$$K_r(x) = K((x - 2^{-r}, x + 2^{-r}) \cap \mathbb{Q}).$$

Lemma 4.2.1. *There is a constant $c \in \mathbb{N}$ such that, for all $S, T \in \Sigma^\infty$ and $r \in \mathbb{N}$,*

$$K((S, T) \upharpoonright r) = K_r(\alpha_S, \alpha_T) + o(r).$$

Proof. First we show that $K_r(\alpha_S, \alpha_T) \leq K((S, T) \upharpoonright r) + o(r)$.

Observe that

$$\begin{aligned} & |(\alpha_S, \alpha_T) - (\alpha_{S \upharpoonright r}, \alpha_{T \upharpoonright r})| \\ &= \left| \left(\sum_{i=0}^{\infty} S[i]k^{-(i+1)}, \sum_{i=0}^{\infty} T[i]k^{-(i+1)} \right) - \left(\sum_{i=0}^{r-1} S[i]k^{-(i+1)}, \sum_{i=0}^{r-1} T[i]k^{-(i+1)} \right) \right| \\ &= \left| \left(\sum_{i=r}^{\infty} S[i]k^{-(i+1)}, \sum_{i=r}^{\infty} T[i]k^{-(i+1)} \right) \right| \\ &\leq \left| \left(\sum_{i=r}^{\infty} S[i]2^{-(i+1)}, \sum_{i=r}^{\infty} T[i]2^{-(i+1)} \right) \right| \\ &= |(2^{-r}, 2^{-r})| \\ &\leq 2^{1-r}, \end{aligned}$$

which implies the inequality

$$K_{r-1}(\alpha_S, \alpha_T) \leq K(\alpha_{S \upharpoonright r}, \alpha_{T \upharpoonright r}). \quad (4.2.1)$$

Let M be a Turing machine such that, if $U(\pi) = (u_0, w_0)(u_1, w_1) \cdots (u_{n-1}, w_{n-1}) \in (\Sigma \times \Sigma)^*$,

$$M(\pi) = \left(\sum_{i=0}^{n-1} u_i \cdot k^{-(i+1)}, \sum_{i=0}^{n-1} w_i \cdot k^{-(i+1)} \right). \quad (4.2.2)$$

Let c_M be an optimality constant for M and $\pi \in \{0, 1\}^*$ be a minimum-length program for $(S, T) \upharpoonright r$. By optimality and (4.2.2),

$$\begin{aligned} K(\alpha_{S \upharpoonright r}, \alpha_{T \upharpoonright r}) &\leq K_M(\alpha_{S \upharpoonright r}, \alpha_{T \upharpoonright r}) \\ &\leq |\pi| + c_M \\ &= K((S, T) \upharpoonright r) + c_M. \end{aligned} \quad (4.2.3)$$

Therefore, by Corollary 2.5.3, (4.2.1), and (4.2.3),

$$\begin{aligned} K_r(\alpha_S, \alpha_T) &\leq K_{r-1}(\alpha_S, \alpha_T) + o(r) \\ &\leq K(\alpha_{S \upharpoonright r}, \alpha_{T \upharpoonright r}) + o(r) \\ &\leq K((S, T) \upharpoonright r) + o(r). \end{aligned}$$

Next we prove that $K((S, T) \upharpoonright r) \leq K_r(\alpha_S, \alpha_T) + O(1)$. We consider the case where $S = x(k-1)^\infty$, $T \neq y(k-1)^\infty$, and $x \in \Sigma^*$ and $y \in \Sigma^*$ are either empty or end with a symbol other than $(k-1)$, i.e., S has a tail that is an infinite sequence of the largest symbol in Σ and T does not. Let M' be a Turing machine such that, if $U(\pi) = \langle q, p \rangle$ for any two rationals $q, p \in [0, 1]$,

$$M'(\pi) = (u_0, w_0)(u_1, w_1) \cdots (u_{r-1}, w_{r-1}) \in (\Sigma \times \Sigma)^*, \quad (4.2.4)$$

where M' operates by running π on U to obtain (q, p) and searching for strings $u = u_0u_1 \cdots u_{r-1}$ and $w = w_0w_1 \cdots w_{r-1}$ such that

$$q = \sum_{i=0}^{|x|-1} u_i k^{-(i+1)} + (k-1)k^{-(|x|+1)}, u_{|x|-1} < (k-1), \text{ and } u_i = (k-1) \text{ for } i \geq |x|, \quad (4.2.5)$$

and

$$w_i \cdot k^{-(i+1)} \leq p - (w_0 \cdot k^{-1} + w_1 \cdot k^{-2} + \cdots + w_{i-1} \cdot k^{-i}) < (w_i + 1) \cdot k^{-(i+1)} \quad (4.2.6)$$

for $0 \leq i < r$.

Let $c_{M'}$ be an optimality constant for M' and $m, t \in \mathbb{N}$ such that $m, t \leq k^r - 1$ and

$$(\alpha_S, \alpha_T) \in [m \cdot k^{-r}, (m+1) \cdot k^{-r}) \times [t \cdot k^{-r}, (t+1) \cdot k^{-r}). \quad (4.2.7)$$

Let

$$(q, p) \in B_{k^{-r}}(\alpha_S, \alpha_T) \cap [m \cdot k^{-r}, (m+1) \cdot k^{-r}) \times [t \cdot k^{-r}, (t+1) \cdot k^{-r}) \cap \mathbb{Q}^2, \quad (4.2.8)$$

and let π be a minimum-length program for (q, p) . First we show that $u_i = S[i]$ for all $0 \leq i < r$. We do not need to consider the case where $i \geq |x|$ because (4.2.5) assures us

that $u_i = S[i]$. Thus we will always assume that $i < |x|$. If $u_0 \neq S[0]$, then, by (4.2.5),

$$q \notin [S[0] \cdot k^{-1}, (S[0] + 1) \cdot k^{-1}).$$

By (4.2.7), this implies that

$$q \notin [m \cdot k^{-r}, (m + 1) \cdot k^{-r}),$$

which contradicts (4.2.8). Now assume that $u_n = S[n]$ for all $n \leq i < r - 1$. If $u_{i+1} \neq S[i + 1]$, then, by (4.2.5),

$$q \notin \left[\sum_{n=0}^i S[n] \cdot k^{-(i+1)} + S[i + 1] \cdot k^{-(i+2)}, \sum_{n=0}^i S[n] \cdot k^{-(i+1)} + (S[i + 1] + 1) \cdot k^{-(i+2)} \right).$$

By (4.2.7), this implies that

$$q \notin [m \cdot k^{-r}, (m + 1) \cdot k^{-r}),$$

which contradicts (4.2.8). Therefore, $u_i = S[i]$ for all $0 \leq i < r$. A similar argument shows that $w_i = T[i]$, so we conclude that $M'(q, p) = (S, T) \upharpoonright r$.

By optimality, (4.2.4), and (4.2.8),

$$\begin{aligned} K((S, T) \upharpoonright r) &\leq K_{M'}((S, T) \upharpoonright r) + c_{M'} \\ &\leq |\pi| + c_{M'} \\ &= K(q, p) + c_{M'} \\ &= K(B_{2^{-r}}(\alpha_S, \alpha_T) \cap [0, 1]^2) + c_{M'} \\ &\leq K_r(\alpha_S, \alpha_T) + O(1), \end{aligned}$$

where the last inequality holds simply because we can design a Turing machine to transform any point from outside the unit square to its edge. All other cases for S and T can be proved in a similar manner. \square

Lemma 4.2.2. *There is a constant $c \in \mathbb{N}$ such that, for all $S \in \Sigma^\infty$ and $r \in \mathbb{N}$,*

$$K(S \upharpoonright r) = K_r(\alpha_S) + c.$$

Proof. Let 0^∞ represent the sequence containing all 0's. It is clear that there exist constants $c_1, c_2 \in \mathbb{N}$ such that

$$K(S \upharpoonright r) = K((S, 0^\infty) \upharpoonright r) + c_1$$

and

$$K_r(\alpha_S, 0) = K_r(\alpha_S) + c_2.$$

Therefore, by the above inequalities and Lemma 4.2.1,

$$\begin{aligned} K(S \upharpoonright r) &= K((S, 0^\infty) \upharpoonright r) + c_1 \\ &= K(\alpha_S, 0) + o(r) + c_1 \\ &= K_r(\alpha_S) + o(r) + c_1 + c_2 \\ &= K_r(\alpha_S) + o(r). \end{aligned} \quad \square$$

4.3 Relating the Dimensions of Sequences to the Dimensions of Reals

We now describe how the dimensions of sequences and the dimensions of reals correspond to one another. First, we state the definitions of the lower and upper dimensions of a sequence.

Definition. The *lower* and *upper dimensions* of $S \in \Sigma^\infty$ are

$$\dim(S) = \liminf_{u \rightarrow S} \frac{K(u)}{|u| \log |\Sigma|}$$

and

$$\text{Dim}(S) = \limsup_{u \rightarrow S} \frac{K(u)}{|u| \log |\Sigma|},$$

respectively.

Next, we recall the definitions of the lower and upper dimensions of a real.

Definition. For any point $x \in \mathbb{R}$, the *lower* and *upper dimensions* of x are

$$\dim(x) = \liminf_{r \rightarrow \infty} \frac{K_r(x)}{r}$$

and

$$\text{Dim}(x) = \limsup_{r \rightarrow \infty} \frac{K_r(x)}{r},$$

respectively.

The next two corollaries describe principles that relate the dimensions of sequences to the dimensions of the sequences' real representations. The first follows from Lemma 4.2.1 and the second follows from Lemma 4.2.2.

Corollary 4.3.1. For all $S, T \in \Sigma^\infty$,

$$\dim(S, T) = \dim(\alpha_S, \alpha_T) \text{ and } \text{Dim}(S, T) = \text{Dim}(\alpha_S, \alpha_T).$$

Corollary 4.3.2. For any sequence $S \in \Sigma^\infty$,

$$\dim(S) = \dim(\alpha_S).$$

4.4 Relating the Mutual Information between Sequences to the Mutual Information between Reals

We now proceed to show that the algorithmic mutual information between the first r bits of S and the first r bits of T is equal to the algorithmic mutual information between the real representation of S and the real representation of T at precision r .

Lemma 4.4.1. There is a constant $c \in \mathbb{N}$ such that, for all $x, y \in \Sigma^*$,

$$K(y | x) \leq K(y | \langle x, K(x) \rangle) + K(K(x)) + c.$$

Proof. Let M be a Turing machine such that, if $U(\pi_1) = K(x)$ and $U(\pi_2, \langle x, K(x) \rangle) = y$,

$$M(\pi_1\pi_2, x) = y.$$

Let $c_M \in \mathbb{N}$ be an optimality constant of M . Assume the hypothesis, and let π_1 be a minimum-length program for $K(x)$ and π_2 be a minimum-length program for y given x and $K(x)$. By optimality,

$$\begin{aligned} K(y | x) &\leq K_M(y | x) + c_M \\ &\leq |\pi_1\pi_2| + c_M \\ &= K(y | \langle x, K(x) \rangle) + K(K(x)) + c, \end{aligned}$$

where $c = c_M$. □

Lemma 4.4.2. *For all $x \in \Sigma^*$, $K(K(x)) = o(|x|)$ as $|x| \rightarrow \infty$.*

Proof. There exist constants $c_1, c_2 \in \mathbb{N}$ such that

$$\begin{aligned} K(K(x)) &\leq \log K(x) + c_1 \\ &\leq \log(|x| + c_2) + c_1 \\ &= o(|x|). \end{aligned}$$

as $|x| \rightarrow \infty$. □

The following lemma is well-known and can be found in [42].

Lemma 4.4.3. *There is a constant $c \in \mathbb{N}$ such that, for all $x, y \in \Sigma^*$,*

$$K(x, y) = K(x) + K(y | x, K(x)) + c.$$

The following is a corollary of Lemma 4.4.3.

Corollary 4.4.4. *There is a constant $c \in \mathbb{N}$ such that, for all $x, y \in \Sigma^*$,*

$$K(x, y) \leq K(x) + K(y | x) + c.$$

Lemma 4.4.5. For all $x, y \in \Sigma^*$,

$$K(y|x) + K(x) \leq K(x, y) + o(|x|) \text{ as } |x| \rightarrow \infty.$$

Proof. By Lemma 4.4.1, there is a constant $c_1 \in \mathbb{N}$ such that

$$K(y|x) \leq K(y|\langle x, K(x) \rangle) + K(K(x)) + c_1.$$

This implies that

$$K(y|x) + K(x) \leq K(y|\langle x, K(x) \rangle) + K(K(x)) + K(x) + c_1.$$

By Lemma 4.4.3, there is a constant $c_2 \in \mathbb{N}$ such that

$$K(y|x) + K(x) \leq K(x, y) + K(K(x)) + c_1 + c_2.$$

Therefore, by Lemma 4.4.2,

$$K(y|x) + K(x) \leq K(x, y) + o(|x|).$$

as $|x| \rightarrow \infty$. □

The rest of this section is about mutual information. We remind the reader of the definition of the mutual information between strings as defined in [42].

Definition. The (*algorithmic*) *mutual information* between $u \in \Sigma^*$ and $w \in \Sigma^*$ is

$$I(x : y) = K(y) - K(y|x).$$

Lemma 4.4.6. For all strings $x, y \in \Sigma^*$,

$$I(x : y) = K(x) + K(y) - K(x, y) + o(|x|).$$

Proof. By definition of mutual information and Lemma 4.4.5,

$$\begin{aligned} I(x : y) &= K(y) - K(y|x) \\ &\geq K(x) + K(y) - K(x, y) + o(|x|). \end{aligned}$$

as $|x| \rightarrow \infty$. Also, by Corollary 4.4.4, there is a constant $c \in \mathbb{N}$ such that

$$\begin{aligned} I(x : y) &= K(y) - K(y | x) \\ &\leq K(x) + K(y) - K(x, y) + c \\ &= K(x) + K(y) - K(x, y) + o(|x|). \end{aligned}$$

as $|x| \rightarrow \infty$. □

The next definition was proposed and thoroughly investigated in Chapter 2.

Definition. The *mutual information* between $x \in \mathbb{R}$ and $y \in \mathbb{R}$ at *precision* $r \in \mathbb{N}$ is

$$I_r(x : y) = \min\{I(q : p) \mid q \in B_{2^{-r}}(x) \cap \mathbb{Q} \text{ and } p \in B_{2^{-r}}(y) \cap \mathbb{Q}\}.$$

Lemma 4.4.7. For all $S, T \in \Sigma^\infty$ and $r \in \mathbb{N}$,

$$I(S \upharpoonright r : T \upharpoonright r) = I_r(\alpha_S : \alpha_T) + o(r).$$

Proof. By Lemmas 4.2.2, 4.2.1, and 4.4.6,

$$\begin{aligned} I(S \upharpoonright r : T \upharpoonright r) &= K(S \upharpoonright r) + K(T \upharpoonright r) - K((S, T) \upharpoonright r) + o(r) \\ &= K_r(\alpha_S) + K_r(\alpha_T) - K_r(\alpha_S, \alpha_T) + o(r) \\ &= I_r(\alpha_S : \alpha_T) + o(r). \end{aligned}$$

as $r \rightarrow \infty$. □

4.5 Relating the Mutual Dimensions between Sequences to the Mutual Dimensions between Reals

In this section, we define the upper and lower mutual dimensions between sequences, recall the definitions of the upper and lower mutual dimensions between reals, and describe how these definitions relate to each other.

Definition. The *lower* and *upper mutual dimensions* between $S \in \Sigma^\infty$ and $T \in \Sigma^\infty$ are

$$mdim(S : T) = \liminf_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{|u| \log |\Sigma|}$$

and

$$Mdim(S : T) = \limsup_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{|u| \log |\Sigma|},$$

respectively.

(We insist that $|u| = |w|$ in the above limits.) The mutual dimension between two sequences is regarded as the density of algorithmic mutual information between them.

Definition. The *lower* and *upper mutual dimensions* between $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are

$$mdim(x : y) = \liminf_{r \rightarrow \infty} \frac{I_r(x : y)}{r}$$

and

$$Mdim(x : y) = \limsup_{r \rightarrow \infty} \frac{I_r(x : y)}{r}$$

The following corollary follows immediately from Lemma 4.4.7 and relates the mutual dimension between sequences to the mutual dimension between the sequences' real representations.

Corollary 4.5.1. For all $S, T \in \Sigma^\infty$,

$$mdim(S : T) = mdim(\alpha_S : \alpha_T) \text{ and } MDim(S : T) = Mdim(\alpha_S : \alpha_T).$$

4.6 Properties of Mutual Dimensions between Sequences

This section describes the basic properties of the lower and upper mutual dimensions between sequences.

Theorem 4.6.1. For all $S, T \in \Sigma^\infty$,

1. $\dim(S) + \dim(T) - \text{Dim}(S, T) \leq \text{mdim}(S : T) \leq \text{Dim}(S) + \text{Dim}(T) - \text{Dim}(S, T)$.
2. $\dim(S) + \dim(T) - \dim(S, T) \leq \text{Mdim}(S : T) \leq \text{Dim}(S) + \text{Dim}(T) - \dim(S, T)$.
3. $\text{mdim}(S : T) \leq \min\{\dim(S), \dim(T)\}$; $\text{Mdim}(S : T) \leq \min\{\text{Dim}(S), \text{Dim}(T)\}$.
4. $0 \leq \text{mdim}(S : T) \leq \text{Mdim}(S : T) \leq 1$.
5. $\text{mdim}(S : T) = \text{mdim}(T : S)$; $\text{Mdim}(S : T) = \text{Mdim}(T : S)$.

Proof. The theorem follows directly from the properties of mutual dimension between points in Euclidean space described in section 3.1 and the correspondences described in corollaries 4.3.1, 4.3.2, and 4.5.1. □

4.7 Turing Reductions and Functionals

Oracle machines are used as a means of carrying out *relative* computations, i.e., computations performed by Turing machines with access to an additional source of information provided by the oracle. An oracle machine is a Turing machine equipped with an additional read-only tape called the *oracle tape*. We write M^S to denote an oracle machine with sequence S written on its oracle tape. Given an input $n \in \mathbb{N}$, an oracle machine will either halt or run forever. If the oracle machine halts on a given input, then it must query the oracle tape a finite number of times.

It is often useful to provide an oracle tape with a string rather than a sequence. The behavior of a machine M with a string oracle $x \in \Sigma^*$ is identical to that of a sequence oracle $S \in \Sigma^\infty$, except that, if the machine attempts to query a position of the oracle tape that is larger than $|x| - 1$, the machine immediately enters a looping state and runs forever.

The following notations and definitions can be found in [1, 57, 64]. We may disassociate an oracle machine M from any particular oracle and refer to it as a partial

function $\Phi_M : \Sigma^\infty \times \mathbb{N} \rightarrow \Sigma^*$ defined by $\Phi_M(S, n) = M^S(n)$. Each Φ_M is called a *Turing functional*. The partial function $\Phi_M^S : \mathbb{N} \rightarrow \Sigma^*$ is defined by $\Phi_M^S(n) = \Phi_M(S, n)$, and we write $\Phi_M^S(n) \downarrow$ if M^S halts on input n and $\Phi_M^S(n) \uparrow$ if M^S does not halt on input n .

For any two sequences S and T and any oracle machine M , we write $\Phi_M^S = T$ if, for all $n \in \mathbb{N}$,

$$\Phi_M^S(n) = T \upharpoonright n.$$

We say that T is *Turing reducible to S* if there exists an oracle machine M such that $\Phi_M^S = T$.

For the rest of this paper, we omit the M in Φ_M and Φ_M^S and denote an arbitrary Turing functional by Φ and an arbitrary Turing functional with oracle S by Φ^S .

4.8 Turing Functionals with Bounded Use and Data

Processing Inequalities

In this section, we develop data processing inequalities for sequences and show how these inequalities change when applying different computable bounds to the *use* of a Turing functional. First, we prove several supporting lemmas.

Lemma 4.8.1. *There exists a constant $c \in \mathbb{N}$ such that, for all $u, v, w \in \Sigma^*$,*

$$K(u \mid vw) \leq K(u \mid v) + K(|v|) + c.$$

Proof. Let M be a TM such that, if $U(\pi_1) = |v|$ and $U(\pi_2, v) = u$,

$$M(\pi_1\pi_2, vw) = u.$$

Let $c_M \in \mathbb{N}$ be an optimality constant of M . Assume the hypothesis, and let π_1 be a minimum-length program for $|v|$ and π_2 be a minimum-length program for u given v . By

optimality,

$$\begin{aligned}
 K(u | vw) &\leq K_M(u | vw) + c_M \\
 &\leq |\pi_1 \pi_2| + c_M \\
 &= K(u | v) + K(|v|) + c,
 \end{aligned}$$

where $c = c_M$. □

Corollary 4.8.2. For all $u, v, w \in \Sigma^*$,

$$I(u : w) \leq I(uv : w) + o(|u|).$$

Proof. By the definition of mutual information and Lemma 4.8.1, there exists a constant $c \in \mathbb{N}$ such that

$$\begin{aligned}
 I(u : w) &= K(w) - K(w | u) \\
 &\leq K(w) - K(w | uv) + K(|u|) + c \\
 &= I(uv : w) + o(|u|).
 \end{aligned}$$
□

Corollary 4.8.3. For all $u, w \in \Sigma^*$,

$$I(u : w) = I(w : u) + o(|u|) + o(|w|).$$

Proof. By Lemma 4.4.6,

$$\begin{aligned}
 I(u : w) &= K(u) + K(w) - K(u, w) + o(|u|) \\
 &= K(w) + K(u) - K(w, u) + o(|u|) \\
 &= I(w : u) + o(|u|) + o(|w|).
 \end{aligned}$$
□

We now investigate bounded Turing reductions and their effects on the shared algorithmic information between strings. As previously mentioned, a halting oracle machine computation can only make a finite number of queries to its oracle, and we are often interested in knowing the largest position of the oracle tape that a machine will query before it halts. The following definition is from [1].

Definition. The *use function* of a Turing functional Φ equipped with oracle $S \in \Sigma^\infty$ is

$$\phi_{use}^S(n) = \begin{cases} m+1 & \text{if } \Phi^S(n) \downarrow \text{ and } m \text{ is the largest query made to } S \\ 0 & \text{if } \Phi^S(n) \downarrow \text{ and } S \text{ is not queried during the computation} \\ \text{undefined} & \text{if } \Phi^S(n) \uparrow \end{cases}$$

for every $n \in \mathbb{N}$.

Traditionally, we denote Turing functionals using uppercase Greek letters (e.g., Φ, Γ) and their corresponding use functions by lowercase Greek letters (e.g., ϕ_{use}, γ_{use}).

Definition. A sequence $T \in \Sigma^\infty$ is *bounded Turing reducible* (*bT-reducible*) to a sequence $S \in \Sigma^\infty$ if T is Turing reducible to S by a Turing functional Φ such that ϕ_{use}^S is bounded by a computable function.

For convenience, we say that $T \in \Sigma^\infty$ is *m-bT-reducible* to $S \in \Sigma^\infty$ if T is bT-reducible to S via Φ and $m : \mathbb{N} \rightarrow \mathbb{N}$ is a computable function bounding ϕ_{use}^S .

Lemma 4.8.4. *Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be a computable, strictly increasing function. For all $X, Y, Z \in \Sigma^\infty$, if Z is m-bT-Turing reducible to X , then*

$$I(Z \upharpoonright r : Y \upharpoonright r) \leq I(X \upharpoonright m(r) : Y \upharpoonright m(r)) + o(m(r)).$$

Proof. Assume that Z is m-bT-Turing reducible to X by some Turing functional Φ whose use function ϕ_{use}^X is bounded by m . By Corollaries 4.8.2 and 4.8.3,

$$\begin{aligned} I(Z \upharpoonright r : Y \upharpoonright r) &= I(Y \upharpoonright r : Z \upharpoonright r) + o(r) \\ &\leq I(Y \upharpoonright m(r) : Z \upharpoonright r) + o(r) \\ &= I(Z \upharpoonright r : Y \upharpoonright m(r)) + o(m(r)). \end{aligned} \tag{4.8.1}$$

Define the partial function $f : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^*$ by

$$f(u, r) = \Phi^u(r),$$

for all $u \in \Sigma^*$ and $r \in \mathbb{N}$. The function f is clearly computable. Therefore, by (4.8.1) and Lemma 3.3.2,

$$\begin{aligned}
I(Z \upharpoonright r : Y \upharpoonright r) &\leq I(f(X \upharpoonright m(r), r) : Y \upharpoonright m(r)) + o(m(r)) \\
&= K(Y \upharpoonright m(r)) - K(Y \upharpoonright m(r) \mid f(X \upharpoonright m(r), r)) + o(m(r)) \\
&\leq K(Y \upharpoonright m(r)) - K(Y \upharpoonright m(r) \mid X \upharpoonright m(r)) + o(m(r)) \\
&= I(X \upharpoonright m(r) : Y \upharpoonright m(r)) + o(m(r)). \quad \square
\end{aligned}$$

We now present an important technical lemma.

Lemma 4.8.5 (Bounded Use Processing Lemma). *Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be a computable, strictly increasing function. For all $X, Y, Z \in \Sigma^\infty$, if Z is m -bT-Turing reducible to X , then*

$$mdim(Z : Y) \leq mdim(X : Y) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right)$$

and

$$Mdim(Z : Y) \leq Mdim(X : Y) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right),$$

except when $\left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right) = \infty$ while either $mdim(X : Y) = 0$ or $Mdim(X : Y) = 0$.

Proof. By Lemma 4.8.4,

$$\begin{aligned}
mdim(Z : Y) &= \liminf_{r \rightarrow \infty} \frac{I(Z \upharpoonright r : Y \upharpoonright r)}{r \log |\Sigma|} \\
&\leq \liminf_{r \rightarrow \infty} \frac{I(X \upharpoonright m(r) : Y \upharpoonright m(r)) + o(m(r))}{r \log |\Sigma|} \\
&= \liminf_{r \rightarrow \infty} \left(\frac{I(X \upharpoonright m(r) : Y \upharpoonright m(r)) + o(m(r))}{m(r) \log |\Sigma|} \cdot \frac{m(r)}{r} \right) \\
&\leq \left(\liminf_{r \rightarrow \infty} \frac{I(X \upharpoonright m(r) : Y \upharpoonright m(r)) + o(m(r))}{m(r) \log |\Sigma|} \right) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right) \\
&= mdim(X : Y) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right).
\end{aligned}$$

A similar proof can be given for $Mdim$. □

Definition. Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $m(n) = n + c$, where $c \in \mathbb{N}$ is a constant. A sequence $T \in \Sigma^\infty$ is *computable Lipschitz reducible (cl-reducible)* to a sequence $S \in \Sigma^\infty$ if T is m -bT-reducible to S .

The following theorem follows directly from Lemma 4.8.5.

Theorem 4.8.6. *For all sequences $X, Y, Z \in \Sigma^\infty$, if Z is cl-reducible to X , then*

$$mdim(Z : Y) \leq mdim(X : Y)$$

and

$$Mdim(Z : Y) \leq Mdim(X : Y).$$

Let $\alpha \geq 1$ and $h_\alpha : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $h_\alpha(n) = \lceil \alpha(n + c) \rceil$, where $c \in \mathbb{N}$ is a constant. The following is a corollary of Lemma 4.8.5.

Corollary 4.8.7. *Let $\alpha \geq 1$. For all sequences $X, Y, Z \in \Sigma^\infty$, if Z is h_α -bT-reducible to a sequence X , then*

$$mdim(Z : Y) \leq \alpha \cdot mdim(X : Y)$$

and

$$Mdim(Z : Y) \leq \alpha \cdot Mdim(X : Y).$$

Typically, data processing inequalities are statements about *all* of the defined outputs of a particular transformation. The results above, while powerful, are not framed in this manner. To remedy this, we now discuss data processing inequalities in terms of individual bounded Turing functionals.

Definition. Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be a computable function. A *m -bounded Turing functional (m -bT-functional)* is a Turing functional such that, for every sequence $S \in \Sigma^\infty$ and every $n \in \mathbb{N}$ where $\Phi^S(n)$ is defined, $\phi_{use}^S(n) \leq m(n)$.

Definition. Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $m(n) = n + c$. A *computable Lipschitz functional (cl-functional)* is a m -bounded Turing functional.

We use Theorem 4.8.6 and Corollary 4.8.7 to derive the following data processing inequalities for sequences whose transformations are bounded Turing functionals.

Corollary 4.8.8. *If Φ is a cl-functional, then, for all $S, T \in \Sigma^\infty$ where Φ^S is defined,*

$$mdim(\Phi^S : T) \leq mdim(S : T)$$

and

$$Mdim(\Phi^S : T) \leq Mdim(S : T).$$

We also have a similar data processing inequality for h_α -bounded Turing functionals.

Corollary 4.8.9. *For all $\alpha \geq 1$, if Φ is a h_α -bounded Turing functional, then, for all $S, T \in \Sigma^\infty$ where Φ^S is defined,*

$$mdim(\Phi^S : T) \leq \alpha \cdot mdim(S : T)$$

and

$$Mdim(\Phi^S : T) \leq \alpha \cdot Mdim(S : T).$$

4.9 Turing Functionals with Bounded Yield and Reverse Data Processing Inequalities

In this section, we define the *yield* of a Turing functional and develop several reverse data processing inequalities (i.e., data processing inequalities where the transformations may significantly *increase* the mutual dimension between two sequences) using yield bounded Turing functionals.

We now introduce the *yield function* of a Turing functional.

Definition. The *yield function* of a Turing functional Φ equipped with oracle $S \in \Sigma^\infty$ is defined by

$$\phi_{yield}^S(n) = \min\{m \in \mathbb{N} \mid \Phi^{S \upharpoonright n}(m) \uparrow\},$$

for all $n \in \mathbb{N}$.

Intuitively, “use” is how much of the oracle the Turing functional must access in order for it to halt on a given input, while “yield” is how many inputs the Turing functional can halt on given a prefix of the oracle.

Definition. A sequence $T \in \Sigma^\infty$ is *yield bounded reducible* (*yb-reducible*) to a sequence $S \in \Sigma^\infty$ if T is Turing reducible to S by a Turing functional Φ such that ϕ_{yield}^S is bounded by a computable function.

For convenience, we say that T is *m-yb-reducible* to S if T is yb-reducible to S and $m : \mathbb{N} \rightarrow \mathbb{N}$ is a computable function bounding ϕ_{yield}^S .

In order to develop reverse data processing inequalities for sequences, we need to apply the following restriction to our Turing functionals.

Definition. A Turing functional Φ^S is *uniquely yielding* for oracle $S \in \Sigma^\infty$ if, for all $T \in \Sigma^\infty$ and $n \in \mathbb{N}$,

$$\Phi^S \upharpoonright \phi_{yield}^S(n) \sqsubseteq \Phi^T \Rightarrow S \upharpoonright n \sqsubseteq T.$$

Definition. A sequence $T \in \Sigma^\infty$ is *uniquely yield bounded reducible* (*uyb-reducible*) to $S \in \Sigma^\infty$ if T is yb-reducible to S by a Turing functional that is uniquely yielding.

We say that T is *m-uyb-reducible* to S if T is uyb-reducible to S by a Turing functional whose yield function is bounded by a computable function $m : \mathbb{N} \rightarrow \mathbb{N}$.

Lemma 4.9.1. *If $T \in \Sigma^\infty$ is m-uyb-reducible to $S \in \Sigma^\infty$, then S is m-bT-reducible to T .*

Proof. Let T be m-uyb-reducible to S by a Turing functional Φ . We define a Turing functional Γ^T that operates on an input $n \in \mathbb{N}$ by querying the first $m(n)$ bits of T and searching for a string $x \in \Sigma^*$ such that $|x| \geq n$ and $\Phi^x(m(n)) = T \upharpoonright m(n)$. After finding

x , Γ^T outputs $x \upharpoonright n$. Observe that

$$\begin{aligned} \Phi^S \upharpoonright \phi_{yield}^S(n) &\sqsubseteq \Phi^S \upharpoonright m(n) \\ &= T \upharpoonright m(n) \\ &= \Phi^x(m(n)) \\ &\sqsubseteq \Phi^x. \end{aligned}$$

Since Φ is uniquely yielding for S and $|x| \geq n$, $S \upharpoonright n \sqsubseteq x$, which implies that $\Gamma^T(n) = S \upharpoonright n$. \square

The following lemma follows directly from Lemma 4.8.5 and Lemma 4.9.1.

Lemma 4.9.2 (Bounded Yield Processing Lemma). *Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be a computable, strictly increasing function. For all $X, Y, Z \in \Sigma^\infty$, if Z is m -uyb-reducible to X , then*

$$mdim(X : Y) \leq mdim(Z : Y) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right)$$

and

$$Mdim(X : Y) \leq Mdim(Z : Y) \left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right),$$

except when $\left(\limsup_{r \rightarrow \infty} \frac{m(r)}{r} \right) = \infty$ while either $mdim(Z : Y) = 0$ or $Mdim(Z : Y) = 0$.

Definition. Let $m(n) = n + c$, for some constant $c \in \mathbb{N}$. A sequence $T \in \Sigma^\infty$ is *linear uniquely yield bounded reducible* (ℓ -uyb-reducible) to a sequence $S \in \Sigma^\infty$ if T is m -uyb-reducible to S .

The following theorem and corollary follow directly from the Bounded Yield Processing Lemma.

Theorem 4.9.3. *For all sequences $X, Y, Z \in \Sigma^\infty$, if Z is ℓ -uyb-reducible to X , then*

$$mdim(X : Y) \leq mdim(Z : Y)$$

and

$$Mdim(X : Y) \leq Mdim(Z : Y).$$

Corollary 4.9.4. *Let $\alpha \geq 1$. For all sequences $X, Y, Z \in \Sigma^\infty$, if Z is h_α -uyb-reducible to X , then*

$$mdim(X : Y) \leq \alpha \cdot mdim(Z : Y)$$

and

$$Mdim(X : Y) \leq \alpha \cdot Mdim(Z : Y).$$

The end of Section 4.8 discussed data processing inequalities in terms of *all* of the defined outputs of use bounded Turing functionals. In like manner, we describe reverse data processing inequalities in terms of yield bounded Turing functionals.

Definition. A Turing functional is a *yield bounded functional* (*yb-functional*) if there exists a computable function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that, for every $S \in \Sigma^\infty$, $\phi_{yield}^S(n) \leq f(n)$.

Definition. A *uniquely yield bounded functional* (*uyb-functional*) is a yield bounded functional that is also uniquely yielding for every oracle.

For convenience, we say that a Turing functional is a *m-uyb-functional* if it is a uyb-functional whose yield is bounded by a computable function $m : \mathbb{N} \rightarrow \mathbb{N}$.

Definition. Let $m : \mathbb{N} \rightarrow \mathbb{N}$ be defined by $m(n) = n + c$. A Turing functional is a *linear uniquely yield bounded functional* (*l-uyb-functional*) if it is a *m-uyb-functional*.

We use Theorem 4.9.3 and Corollary 4.9.4 to derive the following reverse data processing inequalities for sequences whose transformations are uniquely yield bounded Turing functionals.

Corollary 4.9.5. *For all l-uyb-functionals Φ and sequences $S, T \in \Sigma^\infty$ where Φ^S is defined,*

$$mdim(S : T) \leq mdim(\Phi^S : T)$$

and

$$Mdim(S : T) \leq Mdim(\Phi^S : T).$$

Corollary 4.9.6. *Let $\alpha \geq 1$. For all h_α -uyb-functionals Φ and sequences $S, T \in \Sigma^\infty$ where Φ^S is defined,*

$$mdim(S : T) \leq \alpha \cdot mdim(\Phi^S : T)$$

and

$$Mdim(S : T) \leq \alpha \cdot Mdim(\Phi^S : T).$$

CHAPTER 5. COUPLED RANDOMNESS

In this chapter, we investigate the mutual dimensions between coupled random sequences. Intuitively, a coupled random sequence is the interleave of two sequences R_1 and R_2 that are generated by independent tosses of coins whose biases may or may not be correlated. We prove that an interesting class of coupled random sequences can be characterized by Shannon mutual information.

We also show that every independently random pair of sequences has zero mutual dimension. However, we demonstrate that the converse is not true by constructing two sequences that are not independently random and yet have zero mutual dimension.

Finally, we develop a “mutual” version of constructive Billingsley dimension, i.e., constructive dimension with respect to nonuniform probability measures [47], and prove a divergence formula that characterizes the Billingsley mutual dimension between certain kinds of coupled random sequences.

Because coupled randomness is new to algorithmic information theory, the first three sections of this chapter review the technical framework for it.

This chapter is a joint work with Jack H. Lutz and can be found in [9].

5.1 Probability Measures on Alphabets and Sequences

Let Σ be a finite alphabet. A (*Borel*) *probability measure* on the Cantor space Σ^∞ of all infinite sequences over Σ is (conveniently represented by) a function $\nu : \Sigma^* \rightarrow [0, 1]$ with the following two properties.

1. $\nu(\lambda) = 1$, where λ is the empty string.
2. For every $w \in \Sigma^*$, $\nu(w) = \sum_{a \in \Sigma} \nu(wa)$.

Intuitively, here, $\nu(w)$ is the probability that $w \sqsubseteq S$ (w is a *prefix* of S) when $S \in \Sigma^\infty$ is “chosen according to” the probability measure ν .

Most of this chapter concerns a very special class of probability measures on Σ^∞ . For each $n \in \mathbb{N}$, let $\alpha^{(n)}$ be a probability measure on Σ , i.e., $\alpha^{(n)} : \Sigma \rightarrow [0, 1]$, with

$$\sum_{a \in \Sigma} \alpha^{(n)}(a) = 1,$$

and let $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)}, \dots)$ be the sequence of these probability measures on Σ . Then the *product* of $\vec{\alpha}$ (or, emphatically distinguishing it from the products $\nu_1 \times \nu_2$ below, the *longitudinal product* of $\vec{\alpha}$) is the probability measure $\mu[\vec{\alpha}]$ on Σ^∞ defined by

$$\mu[\vec{\alpha}](w) = \prod_{n=0}^{|w|-1} \alpha^{(n)}(w[n])$$

for all $w \in \Sigma^*$, where $w[n]$ is the n^{th} symbol in w . Intuitively, a sequence $S \in \Sigma^\infty$ is “chosen according to” $\mu[\vec{\alpha}]$ by performing the successive experiments $\alpha^{(0)}, \alpha^{(1)}, \dots$ *independently*.

5.2 Coupled Probability Measures

To extend probability to pairs of sequences, we regard $\Sigma \times \Sigma$ as an alphabet and rely on the natural identification between $\Sigma^\infty \times \Sigma^\infty$ and $(\Sigma \times \Sigma)^\infty$. A probability measure on $\Sigma^\infty \times \Sigma^\infty$ is thus a function $\nu : (\Sigma \times \Sigma)^* \rightarrow [0, 1]$. It is convenient to write elements of $(\Sigma \times \Sigma)^*$ as ordered pairs (u, v) , where $u, v \in \Sigma^*$ *have the same length*. With this notation, condition 2 above says that, for every $(u, v) \in (\Sigma \times \Sigma)^*$,

$$\nu(u, v) = \sum_{a, b \in \Sigma} \nu(ua, vb).$$

If ν is a probability measure on $\Sigma^\infty \times \Sigma^\infty$, then the first and second *marginal probability measures* of ν (briefly, the first and second *marginals* of ν) are the functions $\nu_1, \nu_2 : \Sigma^* \rightarrow [0, 1]$ defined by

$$\nu_1(u) = \sum_{v \in \Sigma^{|u|}} \nu(u, v), \quad \nu_2(v) = \sum_{u \in \Sigma^{|v|}} \nu(u, v).$$

It is easy to verify that ν_1 and ν_2 are probability measures on Σ^* . The probability measure ν here is often called a *joint probability measure* on $\Sigma^\infty \times \Sigma^\infty$, or a *coupling* of the probability measures ν_1 and ν_2 .

If ν_1 and ν_2 are probability measures on Σ^∞ , then the *product probability measure* $\nu_1 \times \nu_2$ on $\Sigma^\infty \times \Sigma^\infty$ is defined by

$$(\nu_1 \times \nu_2)(u, v) = \nu_1(u)\nu_2(v)$$

for all $u, v \in \Sigma^*$ with $|u| = |v|$. It is well known and easy to see that $\nu_1 \times \nu_2$ is, indeed a probability measure on $\Sigma^\infty \times \Sigma^\infty$ and that the marginals of $\nu_1 \times \nu_2$ are ν_1 and ν_2 . Intuitively, $\nu_1 \times \nu_2$ is the coupling of ν_1 and ν_2 in which ν_1 and ν_2 are *independent*, or *uncoupled*.

We are most concerned here with coupled longitudinal product probability measures on $\Sigma^\infty \times \Sigma^\infty$. For each $n \in \mathbb{N}$, let $\alpha^{(n)}$ be a probability measure on $\Sigma \times \Sigma$, i.e., $\alpha^{(n)} : \Sigma \times \Sigma \rightarrow [0, 1]$, with

$$\sum_{a, b \in \Sigma} \alpha^{(n)}(a, b) = 1,$$

and let $\vec{\alpha} = (\alpha^{(0)}, \alpha^{(1)}, \dots)$ be the sequence of these probability measures. Then the longitudinal product $\mu[\vec{\alpha}]$ is defined as above, but now treating $\Sigma \times \Sigma$ as the alphabet. It is easy to see that the marginals of $\mu[\vec{\alpha}]$ are $\mu[\vec{\alpha}]_1 = \mu[\vec{\alpha}_1]$ and $\mu[\vec{\alpha}]_2 = \mu[\vec{\alpha}_2]$, where each $\alpha_i^{(n)}$ is the marginal on Σ given by

$$\alpha_1^{(n)}(a) = \sum_{b \in \Sigma} \alpha^{(n)}(a, b), \quad \alpha_2^{(n)}(b) = \sum_{a \in \Sigma} \alpha^{(n)}(a, b).$$

The following class of examples is useful [55] and instructive.

Example 5.2.1. Let $\Sigma = \{0, 1\}$. For each $n \in \mathbb{N}$, fix a real number $\rho_n \in [-1, 1]$, and define the probability measure $\alpha^{(n)}$ on $\Sigma \times \Sigma$ by $\alpha^{(n)}(0, 0) = \alpha^{(n)}(1, 1) = \frac{1+\rho_n}{4}$ and $\alpha^{(n)}(0, 1) = \alpha^{(n)}(1, 0) = \frac{1-\rho_n}{4}$. Then, writing $\alpha^{\vec{\rho}}$ for $\vec{\alpha}$, the longitudinal product $\mu[\alpha^{\vec{\rho}}]$ is a probability measure on $\mathbf{C} \times \mathbf{C}$. It is routine to check that the marginals of $\mu[\alpha^{\vec{\rho}}]$ are

$$\mu[\alpha^{\vec{\rho}}]_1 = \mu[\alpha^{\vec{\rho}}]_2 = \mu,$$

where $\mu(w) = 2^{-|w|}$ is the uniform probability measure on \mathbf{C} .

5.3 Coupled Random Sequences

It is convenient here to use Schnorr's martingale characterization [59, 58, 61, 42, 54, 17] of the algorithmic randomness notion introduced by Martin-Löf [51]. If ν is a probability measure on Σ^∞ , then a ν -martingale is a function $d : \Sigma^* \rightarrow [0, \infty)$ satisfying $d(w)\nu(w) = \sum_{a \in \Sigma} d(wa)\nu(wa)$ for all $w \in \Sigma^*$. A ν -martingale d succeeds on a sequence $S \in \Sigma^\infty$ if $\limsup_{w \rightarrow S} d(w) = \infty$. A ν -martingale d is *constructive*, or *lower semicomputable*, if there is a computable function $\hat{d} : \Sigma^* \times \mathbb{N} \rightarrow \mathbb{Q} \cap [0, \infty]$ such that $\hat{d}(w, t) \leq \hat{d}(w, t+1)$ holds for all $w \in \Sigma^*$ and $t \in \mathbb{N}$, and $\lim_{t \rightarrow \infty} \hat{d}(w, t) = d(w)$ holds for all $w \in \Sigma^*$. A sequence $R \in \Sigma^\infty$ is *random* with respect to a probability measure ν on Σ^* if no lower semicomputable ν -martingale succeeds on R .

If we once again treat $\Sigma \times \Sigma$ as an alphabet, then the above notions all extend naturally to $\Sigma^\infty \times \Sigma^\infty$. Hence, when we speak of a *coupled pair* (R_1, R_2) of *random sequences*, we are referring to a pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to some probability measure ν on $\Sigma^\infty \times \Sigma^\infty$ that is explicit or implicit in the discussion. An extensively studied special case here is that $R_1, R_2 \in \Sigma^\infty$ are defined to be *independently random* with respect to probability measures ν_1, ν_2 , respectively, on Σ^∞ if (R_1, R_2) is random with respect to the product probability measure $\nu_1 \times \nu_2$ on $\Sigma^\infty \times \Sigma^\infty$.

When there is no possibility of confusion, we use such convenient abbreviations as “random with respect to $\vec{\alpha}$ ” for “random with respect to $\mu[\vec{\alpha}]$.”

A trivial transformation of Martin-Löf tests establishes the following well known fact.

Observation 5.3.1. *If ν is a computable probability measure on $\Sigma^\infty \times \Sigma^\infty$ and $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ is random with respect to ν , then R_1 and R_2 are random with respect to the marginals ν_1 and ν_2 .*

Example 5.3.2. If $\vec{\rho}$ is a computable sequence of reals $\rho_n \in [-1, 1]$, $\alpha^{\vec{\rho}}$ is as in Example 5.2.1, and $(R_1, R_2) \in \mathbf{C} \times \mathbf{C}$ is random with respect to $\alpha^{\vec{\rho}}$, then Observation 5.3.1 tells us that R_1 and R_2 are random with respect to the uniform probability measure on \mathbf{C} .

5.4 Shannon Entropy Characterizations of the Dimensions of Random Sequences

We recall basic definitions from Shannon information theory.

Definition. Let α be a probability measure on Σ . The *Shannon entropy* of α is

$$\mathcal{H}(\alpha) = \sum_{a \in \Sigma} \alpha(a) \log \frac{1}{\alpha(a)}.$$

Definition. Let α be probability measures on $\Sigma \times \Sigma$. The *Shannon mutual information* between α_1 and α_2 is

$$I(\alpha_1 : \alpha_2) = \sum_{(a,b) \in \Sigma \times \Sigma} \alpha(a,b) \log \frac{\alpha(a,b)}{\alpha_1(a)\alpha_2(b)}.$$

Theorem 5.4.1 ([44]). *If $\vec{\alpha}$ is a computable sequence of probability measures $\alpha^{(n)}$ on Σ that converge to a probability measure α on Σ , then for every $R \in \Sigma^\infty$ that is random with respect to $\vec{\alpha}$,*

$$\dim(R) = \frac{\mathcal{H}(\alpha)}{\log |\Sigma|}.$$

5.5 Shannon Mutual Information Characterizations of the Mutual Dimensions of Coupled Random Sequences

The following is a corollary to Theorem 5.4.1.

Corollary 5.5.1. *If $\vec{\alpha}$ is a computable sequence of probability measures $\alpha^{(n)}$ on Σ that converge to a probability measure α on Σ , then for every $R \in \Sigma^\infty$ that is random with respect to $\vec{\alpha}$ and every $w \sqsubseteq R$,*

$$K(w) = |w|\mathcal{H}(\alpha) + o(|w|).$$

Lemma 5.5.2. *If $\vec{\alpha}$ is a computable sequence of probability measures $\alpha^{(n)}$ on $\Sigma \times \Sigma$ that converge to a probability measure α on $\Sigma \times \Sigma$, then for every coupled pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to $\vec{\alpha}$ and $(u, w) \sqsubseteq (R_1, R_2)$,*

$$I(u : w) = |u|I(\alpha_1 : \alpha_2) + o(|u|).$$

Proof. By Lemma 4.4.6,

$$I(u : w) = K(u) + K(w) - K(u, w) + o(|u|).$$

We then apply Observation 5.3.1 and Corollary 5.5.1 to obtain

$$\begin{aligned} I(u : w) &= |u|(\mathcal{H}(\alpha_1) + \mathcal{H}(\alpha_2) - \mathcal{H}(\alpha)) + o(|u|) \\ &= |u|I(\alpha_1 : \alpha_2) + o(|u|). \end{aligned} \quad \square$$

The following is a corollary to Lemma 5.5.2.

Corollary 5.5.3. *If α is a computable, positive probability measure on $\Sigma \times \Sigma$, then, for every sequence $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to α and $(u, w) \sqsubseteq (R_1, R_2)$,*

$$I(u : w) = |u|I(\alpha_1 : \alpha_2) + o(|u|).$$

In applications one often encounters longitudinal product measures $\mu[\vec{\alpha}]$ in which the probability measures $\alpha^{(n)}$ are all the same (the i.i.d. case) or else converge to some limiting probability measure. The following theorem says that, in such cases, the mutual dimensions of coupled pairs of random sequences are easy to compute.

Theorem 5.5.4. *If $\vec{\alpha}$ is a computable sequence of probability measures $\alpha^{(n)}$ on $\Sigma \times \Sigma$ that converge to a probability measure α on $\Sigma \times \Sigma$, then for every coupled pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to $\vec{\alpha}$,*

$$mdim(R_1 : R_2) = Mdim(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\log |\Sigma|}.$$

Proof. By Lemma 5.5.2, we have

$$\begin{aligned} mdim(R_1 : R_2) &= \liminf_{(u,w) \rightarrow (R_1, R_2)} \frac{I(u : w)}{|u| \log |\Sigma|} \\ &= \liminf_{(u,w) \rightarrow (R_1, R_2)} \frac{|u| I(\alpha_1 : \alpha_2) + o(|u|)}{|u| \log |\Sigma|} \\ &= \frac{I(\alpha_1 : \alpha_2)}{\log |\Sigma|} \end{aligned}$$

A similar proof shows that $Mdim(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\log |\Sigma|}$. □

Example 5.5.5. Let $\Sigma = \{0, 1\}$, and let $\vec{\rho}$ be a computable sequence of reals $\rho_n \in [-1, 1]$ that converge to a limit ρ . Define the probability measure α on $\Sigma \times \Sigma$ by $\alpha(0, 0) = \alpha(1, 1) = \frac{1+\rho}{4}$ and $\alpha(0, 1) = \alpha(1, 0) = \frac{1-\rho}{4}$, and let α_1 and α_2 be the marginals of α . If $\alpha^{\vec{\rho}}$ is as in Example 5.2.1, then for every pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to $\alpha^{\vec{\rho}}$, Theorem 5.5.4 tells us that

$$\begin{aligned} mdim(R_1 : R_2) &= Mdim(R_1 : R_2) \\ &= I(\alpha_1 : \alpha_2) \\ &= 1 - \mathcal{H}\left(\frac{1+\rho}{2}\right). \end{aligned}$$

In particular, if the limit ρ is 0, then

$$mdim(R_1 : R_2) = Mdim(R_1 : R_2) = 0.$$

5.6 The Mutual Dimension between Independently Random Sequences

Theorem 5.5.4 has the following easy consequence, which generalizes the last sentence of Example 5.5.5.

Corollary 5.6.1. *If $\vec{\alpha}$ is a computable sequence of probability measures $\alpha^{(n)}$ on $\Sigma \times \Sigma$ that converge to a product probability measure $\alpha_1 \times \alpha_2$ on $\Sigma \times \Sigma$, then for every coupled pair $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to $\vec{\alpha}$,*

$$mdim(R_1 : R_2) = Mdim(R_1 : R_2) = 0.$$

Applying Corollary 5.6.1 to a constant sequence $\vec{\alpha}$ in which each $\alpha^{(n)}$ is a product probability measure $\alpha_1 \times \alpha_2$ on $\Sigma \times \Sigma$ gives the following.

Corollary 5.6.2. *If α_1 and α_2 are computable probability measures on Σ , and if $R_1, R_2 \in \Sigma^\infty$ are independently random with respect to α_1, α_2 , respectively, then*

$$mdim(R_1 : R_2) = Mdim(R_1 : R_2) = 0.$$

5.7 Dependent Sequences with Zero Mutual Dimension

In this section, we show that the converse of Corollary 5.6.2 does *not* hold. This can be done via a direct construction, but it is more instructive to use a beautiful theorem of Kakutani, van Lambalgen, and Vovk. The *Hellinger distance* between two probability measures α_1 and α_2 on Σ is

$$H(\alpha_1, \alpha_2) = \sqrt{\sum_{a \in \Sigma} (\sqrt{\alpha_1(a)} - \sqrt{\alpha_2(a)})^2}.$$

(See [37], for example.) A sequence $\alpha = (\alpha^{(0)}, \alpha^{(1)}, \dots)$ of probability measures on Σ is *strongly positive* if there is a real number $\delta > 0$ such that, for all $n \in \mathbb{N}$ and $a \in \Sigma$, $\alpha^{(n)}(a) \geq \delta$. Kakutani [33] proved the classical, measure-theoretic version of the

following theorem, and van Lambalgen [70, 71] and Vovk [72] extended it to algorithmic randomness.

Theorem 5.7.1. *Let $\vec{\alpha}$ and $\vec{\beta}$ be computable, strongly positive sequences of probability measures on Σ .*

1. *If*

$$\sum_{n=0}^{\infty} H(\alpha^{(n)}, \beta^{(n)})^2 < \infty,$$

then a sequence $R \in \Sigma^\infty$ is random with respect to $\vec{\alpha}$ if and only if it is random with respect to $\vec{\beta}$.

2. *If*

$$\sum_{n=0}^{\infty} H(\alpha^{(n)}, \beta^{(n)})^2 = \infty,$$

then no sequence is random with respect to both $\vec{\alpha}$ and $\vec{\beta}$.

Observation 5.7.2. *Let $\Sigma = \{0, 1\}$. If $\rho = [-1, 1]$ and probability measure α on $\Sigma \times \Sigma$ is defined from ρ as in Example 5.5.5, then*

$$H(\alpha_1 \times \alpha_2, \alpha)^2 = 2 - \sqrt{1 + \rho} - \sqrt{1 - \rho}.$$

Proof. Assume the hypothesis. Then

$$\begin{aligned} H(\alpha_1 \times \alpha_2, \alpha)^2 &= \sum_{a,b \in \{0,1\}} (\sqrt{\alpha_1(a)\alpha_2(b)} - \sqrt{\alpha(a,b)})^2 \\ &= \sum_{a,b \in \{0,1\}} \left(\frac{1}{2} - \sqrt{\alpha(a,b)} \right)^2 \\ &= 2 \left(\frac{1}{2} - \sqrt{\frac{1+\rho}{4}} \right)^2 + 2 \left(\frac{1}{2} - \sqrt{\frac{1-\rho}{4}} \right)^2 \\ &= 2 - \sqrt{1+\rho} - \sqrt{1-\rho}. \end{aligned} \quad \square$$

Corollary 5.7.3. *Let $\Sigma = \{0, 1\}$ and $\delta \in (0, 1)$. Let $\vec{\rho}$ be a computable sequence of real numbers $\rho_n \in [\delta - 1, 1 - \delta]$, and let $\alpha^{\vec{\rho}}$ be as in Example 5.2.1. If*

$$\sum_{n=0}^{\infty} \rho_n^2 = \infty,$$

and if $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ is random with respect to $\alpha^{\vec{\rho}}$, then R_1 and R_2 are not independently random with respect to the uniform probability measure on \mathbf{C} .

Proof. This follows immediately from Theorem 5.7.1, Observation 5.7.2, and the fact that

$$\sqrt{1+x} + \sqrt{1-x} = 2 - \frac{x^2}{2} + o(x^2)$$

as $x \rightarrow 0$. □

Corollary 5.7.4. *There exist sequences $R_1, R_2 \in \mathbf{C}$ that are random with respect to the uniform probability measure on \mathbf{C} and satisfy $Mdim(R_1 : R_2) = 0$, but are not independently random.*

Proof. For each $n \in \mathbb{N}$, let

$$\rho_n = \frac{1}{\sqrt{n+2}}.$$

Let $\vec{\rho} = (\rho_0, \rho_1, \dots)$, let $\alpha^{\vec{\rho}}$ be as in Example 5.2.1, and let $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ be random with respect to $\alpha^{\vec{\rho}}$. Observation 5.3.1 tells us that R_1 and R_2 are random with respect to the marginals of $\alpha^{\vec{\rho}}$, both of which are the uniform probability measure on \mathbf{C} . Since $\rho_n \rightarrow 0$ as $n \rightarrow \infty$, the last sentence in Example 5.5.5 tells us (via Theorem 5.5.4) that $Mdim(R_1 : R_2) = 0$. Since

$$\sum_{n=0}^{\infty} \rho_n^2 = \sum_{n=0}^{\infty} \frac{1}{n+2} = \infty,$$

Corollary 5.7.3 tells us that R_1 and R_2 are not independently random. □

5.8 Billingsley Dimension

In this section, we review the Billingsley generalization of constructive dimension, i.e., dimension with respect to strongly positive probability measures. A probability measure β on Σ^∞ is *strongly positive* if there exists $\delta > 0$ such that, for all $w \in \Sigma^*$ and $a \in \Sigma$, $\beta(wa) > \delta\beta(w)$.

Definition. The *Shannon self-information* of $w \in \Sigma$ is

$$\ell_\beta(w) = \sum_{i=0}^{|w|-1} \log \frac{1}{\beta(w[i])}.$$

In [47], Lutz and Mayordomo defined (and usefully applied) constructive Billingsley dimension in terms of gales and proved that it can be characterized using Kolmogorov complexity. Since Kolmogorov complexity is more relevant in this discussion, we treat the following theorem as a definition.

Definition (Lutz and Mayordomo [47]). The *dimension* of $S \in \Sigma^\infty$ with respect to a strongly positive probability measure β on Σ^∞ is

$$\dim^\beta(S) = \liminf_{w \rightarrow S} \frac{K(w)}{\ell_\beta(w)}.$$

5.9 Billingsley Mutual Dimension

In the definition of the dimension of a sequence with respect to a strongly positive probability measure on Σ^∞ , the denominator $\ell_\beta(w)$ normalizes the dimension to be a real number in $[0, 1]$. It seems natural to define the Billingsley generalization of mutual dimension in a similar way by normalizing the algorithmic mutual information between u and w by $\log \frac{\beta(u,w)}{\beta_1(u)\beta_2(w)}$ (i.e., the *self-mutual information* or *pointwise mutual information* between u and w [27]) as $(u, w) \rightarrow (S, T)$. However, this results in bad behavior. For example, the mutual dimension between any two sequences with respect to the uniform probability measure on $\Sigma \times \Sigma$ is *always* undefined. Other thoughtful modifications to this natural definition results in sequences having negative or infinitely large mutual dimension. The main problem here is that, given a particular probability measure, one can construct certain sequences whose prefixes have extremely large positive or negative self-mutual information. In order to avoid undesirable behavior, we restrict the definition of Billingsley mutual dimension to sequences that are mutually normalizable.

Definition. Let β be a probability measure on $\Sigma^\infty \times \Sigma^\infty$. Two sequences $S, T \in \Sigma^\infty$ are *mutually β -normalizable* (in this order) if

$$\lim_{(u,w) \rightarrow (S,T)} \frac{\ell_{\beta_1}(u)}{\ell_{\beta_2}(w)} = 1.$$

Definition. Let $S, T \in \Sigma^\infty$ be mutually β -normalizable. The *upper* and *lower mutual dimensions* between S and T with respect to β are

$$mdim^\beta(S : T) = \liminf_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{\ell_{\beta_1}(u)} = \liminf_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{\ell_{\beta_2}(w)}$$

and

$$Mdim^\beta(S : T) = \limsup_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{\ell_{\beta_1}(u)} = \limsup_{(u,w) \rightarrow (S,T)} \frac{I(u : w)}{\ell_{\beta_2}(w)},$$

respectively.

The above definition has nice properties because β -normalizable sequences have prefixes with asymptotically equivalent self-information. Given the basic properties of mutual information and Shannon self-information, we can see that

$$0 \leq mdim^\beta(S : T) \leq \min\{dim^{\beta_1}(S), dim^{\beta_2}(T)\} \leq 1.$$

Clearly, $Mdim^\beta$ also has a similar property.

5.10 A Mutual Divergence Formula

Definition. Let α and β be probability measure on Σ . The *Kullback-Leibler divergence* between α and β is

$$\mathcal{D}(\alpha||\beta) = \sum_{a \in \Sigma} \alpha(a) \log \frac{\alpha(a)}{\beta(a)}$$

The following lemma is useful when proving Lemma 5.11.1 and Theorem 5.10.2.

Lemma 5.10.1 (Frequency Divergence Lemma [46]). *If α and β are positive probability measures on Σ , then, for all $S \in \text{FREQ}^\alpha$,*

$$\ell_\beta(w) = (\mathcal{H}(\alpha) + \mathcal{D}(\alpha||\beta))|w| + o(|w|)$$

as $w \rightarrow S$.

The rest of this chapter is primarily concerned with probability measures on alphabets. Our first result of this section is a mutual divergence formula for random, mutually β -normalizable sequences. This can be thought of as a “mutual” version of a divergence formula in [46].

Theorem 5.10.2 (Mutual Divergence Formula). *If α and β are computable, positive probability measures on $\Sigma \times \Sigma$, then, for every $(R_1, R_2) \in \Sigma^\infty \times \Sigma^\infty$ that is random with respect to α such that R_1 and R_2 are mutually β -normalizable,*

$$mdim^\beta(R_1 : R_2) = Mdim^\beta(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 || \beta_1)} = \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_2) + \mathcal{D}(\alpha_2 || \beta_2)}.$$

Proof. By Corollary 5.5.3 and the Frequency Divergence Lemma, we have

$$\begin{aligned} mdim^\beta(R_1 : R_2) &= \liminf_{(u,w) \rightarrow (R_1, R_2)} \frac{I(u : w)}{\ell_{\beta_1}(u)} \\ &= \liminf_{(u,w) \rightarrow (R_1, R_2)} \frac{|u|I(\alpha_1 : \alpha_2) + o(|u| \log |\Sigma|)}{(\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 || \beta_1))|u| + o(|u|)} \\ &= \liminf_{(u,w) \rightarrow (R_1, R_2)} \frac{|u|(I(\alpha_1 : \alpha_2) + o(\log |\Sigma|))}{|u|((\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 || \beta_1)) + o(1))} \\ &= \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 || \beta_1)}. \end{aligned}$$

Similar arguments show that

$$mdim^\beta(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_2) + \mathcal{D}(\alpha_2 || \beta_2)}$$

and

$$Mdim^\beta(R_1 : R_2) = \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 || \beta_1)} = \frac{I(\alpha_1 : \alpha_2)}{\mathcal{H}(\alpha_2) + \mathcal{D}(\alpha_2 || \beta_2)}. \quad \square$$

5.11 Achieving Mutual Normalizability

We conclude this chapter by making some initial observations regarding when mutual normalizability can be achieved.

Definition. Let $\alpha_1, \alpha_2, \beta_1, \beta_2$ be probability measures over Σ . We say that α_1 is (β_1, β_2) -equivalent to α_2 if

$$\sum_{a \in \Sigma} \alpha_1(a) \log \frac{1}{\beta_1(a)} = \sum_{a \in \Sigma} \alpha_2(a) \log \frac{1}{\beta_2(a)}.$$

For a probability measure α on Σ , let $FREQ_\alpha$ be the set of sequences $S \in \Sigma^\infty$ satisfying $\lim_{n \rightarrow \infty} n^{-1} |\{i < n \mid S[i] = a\}| = \alpha(a)$ for all $a \in \Sigma$.

Lemma 5.11.1. *Let $\alpha_1, \alpha_2, \beta_1, \beta_2$ be probability measures on Σ . If α_1 is (β_1, β_2) -equivalent to α_2 , then, for all pairs $(S, T) \in FREQ_{\alpha_1} \times FREQ_{\alpha_2}$, S and T are mutually β -normalizable.*

Proof. By the Frequency Divergence Lemma,

$$\begin{aligned} \lim_{(u,w) \rightarrow (S,T)} \frac{\ell_{\beta_1}(u)}{\ell_{\beta_2}(w)} &= \lim_{n \rightarrow \infty} \frac{(\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 \parallel \beta_1)) \cdot n + o(n)}{(\mathcal{H}(\alpha_2) + \mathcal{D}(\alpha_2 \parallel \beta_2)) \cdot n + o(n)} \\ &= \frac{\mathcal{H}(\alpha_1) + \mathcal{D}(\alpha_1 \parallel \beta_1)}{\mathcal{H}(\alpha_2) + \mathcal{D}(\alpha_2 \parallel \beta_2)} \\ &= \frac{\sum_{a \in \Sigma} \alpha_1(a) \log \frac{1}{\beta_1(a)}}{\sum_{a \in \Sigma} \alpha_2(a) \log \frac{1}{\beta_2(a)}} \\ &= 1, \end{aligned}$$

where the last equality is due to α_1 being (β_1, β_2) -equivalent to α_2 . □

Given probability measures β_1 and β_2 on Σ , we would like to know which sequences are mutually β -normalizable. The following results help to answer this question for probability measures on and sequences over $\{0, 1\}$.

Lemma 5.11.2. *Let β_1 and β_2 be probability measures on $\{0, 1\}$ such that exactly one of the following conditions hold.*

1. $0 < \beta_2(0) < \beta_1(1) < \beta_1(0) < \beta_2(1) < 1$
2. $0 < \beta_2(1) < \beta_1(0) < \beta_1(1) < \beta_2(0) < 1$

$$3. 0 < \beta_2(0) < \beta_1(0) < \beta_1(1) < \beta_2(1) < 1$$

$$4. 0 < \beta_2(1) < \beta_1(1) < \beta_1(0) < \beta_2(0) < 1$$

$$5. \beta_1 = \mu \text{ and } \beta_2 \neq \mu.$$

If f is defined by

$$f(x) = \frac{x \cdot \log \frac{\beta_1(1)}{\beta_1(0)} + \log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}},$$

then

$$0 < f(x) < 1,$$

for all $x \in [0, 1]$.

Proof. First, observe that f is linear and has a negative slope under conditions 1 and 2, a positive slope under conditions 3 and 4, and zero slope under condition 5. We verify that, for all $x \in [0, 1]$, $f(x) \in (0, 1)$ under each condition.

Under condition 1, we assume

$$\beta_2(0) < \beta_1(1) < \beta_2(1),$$

which implies that

$$\log \frac{\beta_2(0)}{\beta_2(1)} < \log \frac{\beta_1(1)}{\beta_2(1)} < 0.$$

From the above inequality, we obtain

$$0 < \frac{\log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}} < 1.$$

Therefore, by the definition of f ,

$$0 < f(0) < 1. \tag{5.11.1}$$

Under the same condition, we have

$$\beta_1(0) < \beta_2(1),$$

which implies that

$$\log \frac{\beta_1(0)}{\beta_1(1)} < \log \frac{\beta_2(1)}{\beta_1(1)}.$$

From the above inequality, we obtain

$$\frac{\log \frac{\beta_1(0)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}} < \frac{\log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}},$$

whence

$$0 < \frac{\log \frac{\beta_1(1)}{\beta_1(0)} + \log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}}.$$

Therefore, by the definition of f ,

$$0 < f(1). \quad (5.11.2)$$

By (5.11.1), (5.11.2), and the negativity of the slope of f ,

$$0 < f(1) < f(0) < 1.$$

A similar argument shows that, if condition 2 holds, then $0 < f(1) < f(0) < 1$.

Assuming condition 3, we can prove that, if $\beta_2(0) < \beta_1(1) < \beta_2(1)$, then

$$0 < f(0) < 1, \quad (5.11.3)$$

using the argument given above. Under the same condition, we have

$$\beta_2(0) < \beta_1(0),$$

which implies that

$$\log \beta_1(1) - \log \beta_1(0) + \log \beta_2(1) - \log \beta_1(1) < \log \beta_2(1) - \log \beta_2(0).$$

From this inequality, we derive

$$\frac{\log \frac{\beta_1(1)}{\beta_1(0)} + \log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}} < 1.$$

Therefore, by the definition of f ,

$$f(1) < 1. \quad (5.11.4)$$

By (5.11.3), (5.11.4), and the positivity of the slope of f ,

$$0 < f(0) < f(1) < 1.$$

A similar argument shows that, if condition 4 holds, then $0 < f(1) < f(0) < 1$.

Under condition 5 and without loss of generality, assume that $\beta_1 = \mu$ and $\beta_2(0) < 1/2 < \beta_2(1)$, which implies

$$0 < 1 + \log \beta_2(1) < \log \frac{\beta_2(1)}{\beta_2(0)}.$$

From the above inequality, we derive

$$0 < \frac{\log \frac{\beta_2(1)}{1/2}}{\log \frac{\beta_2(1)}{\beta_2(0)}} < 1,$$

whence, by the definition of f ,

$$0 < f(x) < 1,$$

for all $x \in [0, 1]$. □

Theorem 5.11.3. *Let β_1 and β_2 be probability measures on $\{0, 1\}$ that satisfy exactly one of the conditions from Lemma 5.11.2, and let α_1 be an arbitrary probability measure on $\{0, 1\}$. Then α_1 is (β_1, β_2) -equivalent to exactly one unique probability measure α_2 , which is defined by*

$$\alpha_2(0) = \frac{\alpha_1(0) \log \frac{\beta_1(1)}{\beta_1(0)} + \log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}} \quad \text{and} \quad \alpha_2(1) = 1 - \alpha_2(0).$$

Proof. By Lemma 5.11.2, α_2 is a valid probability measure. Observe that

$$\alpha_2(0) = \frac{\alpha_1(0) \log \frac{\beta_1(1)}{\beta_1(0)} + \log \frac{\beta_2(1)}{\beta_1(1)}}{\log \frac{\beta_2(1)}{\beta_2(0)}}$$

if and only if

$$\alpha_1(0) \left(\log \frac{1}{\beta_1(0)} - \log \frac{1}{\beta_1(1)} \right) + \log \frac{1}{\beta_1(1)} = \alpha_2(0) \left(\log \frac{1}{\beta_2(0)} - \log \frac{1}{\beta_2(1)} \right) + \log \frac{1}{\beta_2(1)}.$$

The above equality holds if and only if

$$\alpha_1(0) \log \frac{1}{\beta_1(0)} + \alpha_1(1) \log \frac{1}{\beta_1(1)} = \alpha_2(0) \log \frac{1}{\beta_2(0)} + \alpha_2(1) \log \frac{1}{\beta_2(1)},$$

which implies that α_1 is (β_1, β_2) -equivalent to α_2 . □

The following corollary follows from Theorem 5.11.3 and Lemma 5.11.1.

Corollary 5.11.4. *Let β_1 , β_2 , α_1 , and α_2 be as defined in Theorem 5.11.3. For all $(S, T) \in \text{FREQ}_{\alpha_1} \times \text{FREQ}_{\alpha_2}$, S and T are mutually β -normalizable.*

BIBLIOGRAPHY

- [1] Klaus Ambos-Spies. Strongly Bounded Turing Reducibilities and Computably Enumerable Sets. Heidelberg University Lecture Notes, 2011.
- [2] Krishna B. Athreya, John M. Hitchcock, Jack H. Lutz, and Elvira Mayordomo. Effective strong dimension in algorithmic information and computational complexity. *SIAM Journal of Computing*, 37(3):671–705, 2007.
- [3] C.B. Bell. Mutual information and maximal correlation as measures of dependence. *Annals of Mathematical Statistics*, 33(2):587–595, 1962.
- [4] P. Billingsley. Hausdorff dimension in probability theory. *Illinois Journal of Mathematics*, 4:187–209, 1960.
- [5] P. Billingsley. *Ergodic Theory and Information*. R. E. Krieger Pub. Co, 1978.
- [6] Mark Braverman and Stephen Cook. Computing over the reals: foundations for scientific computing. *Notices of the American Mathematical Society*, 53(3):318–329, 2006.
- [7] H. Cajar. *Billingsley Dimension in Probability Spaces*, volume 892 of *Lecture Notes in Mathematics*. Springer, 1981.
- [8] Adam Case. Bounded Turing reductions and data processing inequalities for sequences, submitted.

- [9] Adam Case and Jack H. Lutz. Mutual dimension and random sequences. In *Proceedings of the 40th International Symposium on the Mathematical Foundations of Computer Science*, pages 199–210. Springer, 2015.
- [10] Adam Case and Jack H. Lutz. Mutual dimension. *ACM Transactions on Computation Theory*, 7, July 2015, article no. 12.
- [11] Gregory J. Chaitin. On the simplicity and speed of programs for computing infinite sets of natural numbers. *Journal of the ACM*, 16:407–422, 1969.
- [12] Chris J. Conidis. Effective packing dimension of Π_1^0 -classes. *Proceedings of the American Mathematical Society*, 136:3655–3662, 2008.
- [13] P. J. Couch, B. D. Daniel, and Timothy H. McNicholl. Computing space-filling curves. *Theory of Computing Systems*, 50(2):370–386, 2012.
- [14] Thomas R. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., second edition, 2006.
- [15] Randall Dougherty, Jack H. Lutz, Daniel R. Mauldin, and Jason Teutsch. Translating the Cantor set by a random real. *Transactions of the American Mathematical Society*, 366(6):3027–3041, 2014.
- [16] Rod G. Downey, Denis R. Hirschfeldt, and Geoff LaForte. Randomness and reducibility. *Journal of Computer and System Sciences*, 68:96–114, 2004.
- [17] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010 edition, 2010.
- [18] H. G. Eggleston. The fractional dimension of a set defined by decimal properties. *The Quarterly Journal of Mathematics*, 20:31–36, 1965.

- [19] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, second edition, 2003.
- [20] Lance Fortnow, John M. Hitchcock, A. Pavan, N.V. Vinodchandran, and Fengming Wang. Extracting Kolmogorov complexity with applications to dimension zero-one laws. *Information and Computation*, 209:627–636, 2011.
- [21] Peter Gács. On the symmetry of algorithmic information. *Soviet Mathematics Doklady*, 15(5):1477–1480, 1974. Correction, *Ibid.*, 15:1480, 1974.
- [22] Péter Gács. Every sequence is reducible to a random one. *Information and Control*, 70:186–192, 1986.
- [23] R. Gavaldà, M. López-Valdés, E. Mayordomo, and N.V. Vinodchandran. Resource-bounded dimension in computational learning theory. arXiv:1010.5470, January 2011.
- [24] Xiaoyang Gu, Jack Lutz, and Elvira Mayordomo. Curves that must be retraced. *Information and Computation*, 209(6):992–1006, 2011.
- [25] Xiaoyang Gu, Jack H. Lutz, R. Elvira Mayordomo, and Philippe Moser. Dimension spectra of random subfractals of self-similar fractals. *Annals of Pure and Applied Logic*, 165:1707–1726, 2014.
- [26] Daniel Hammer, Andrei E. Romashchenko, Alexander Shen, and Nikolai K. Vereshchagin. Inequalities for Shannon entropy and Kolmogorov complexity. *Journal of Computer System Sciences*, 60(2):442–464, 2000.
- [27] Te Sun Han and Kingo Kobayashi. *Mathematics of Information and Coding*. Translations of Mathematical Monographs (Book 203). American Mathematical Society, 2007.

- [28] John M. Hitchcock. Gales suffice for constructive dimension. *Information Processing Letters*, 86:9–12, 2003.
- [29] John M. Hitchcock. Correspondence principles for effective dimensions. *Theory of Computing Systems*, 38(5):559–571, 2005.
- [30] John M. Hitchcock. Online learning and resource-bounded dimension: Winnow yields new lower bounds for hard sets. *SIAM Journal on Computing*, 36:1696–1708, 2007.
- [31] John M. Hitchcock, Jack H. Lutz, and Elvira Mayordomo. The fractal geometry of complexity classes. *SIGACT News*, 36:24–38, 2005.
- [32] Claude Tricot Jr. Two definitions of fractional dimension. *Mathematical Proceedings of the Cambridge Philosophical Society*, 91:57–74, 1982.
- [33] Shizuo Kakutani. On equivalence of infinite product measures. *Annals of Mathematics*, 49(1):214–224, 1948.
- [34] Bjørn Kjos-Hanssen and Anil Nerode. Effective dimension of points visited by Brownian motion. *Theoretical Computer Science*, 410(4-5):347–354, 2009.
- [35] Ker-I Ko. *Complexity Theory of Real Functions*. Birkhäuser, first edition, 1991.
- [36] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.
- [37] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, first edition, 2009.
- [38] Leonid A. Levin. On the notion of a random sequence. *Soviet Mathematics Doklady*, 14(5):1413–1416, 1973.

- [39] Leonid A. Levin. Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Problemy Peredachi Informatsii*, 10(3):30–35, 1974.
- [40] Andrew E.M. Lewis and George Barmpalias. Random reals and lipschitz continuity. *Mathematical Structures in Computer Science*, 16:737–749, 2006.
- [41] Andrew E.M. Lewis and George Barmpalias. Randomness and the linear degrees of computability. *Annals of Pure and Applied Logic*, 145:252–257, 2006.
- [42] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, third edition, 2008.
- [43] Jack H. Lutz. Dimension in complexity classes. *SIAM Journal on Computing*, 32(5):1235–1259, 2003.
- [44] Jack H. Lutz. The dimensions of individual strings and sequences. *Information and Computation*, 187(1):49–79, 2003.
- [45] Jack H. Lutz. Effective fractal dimensions. *Mathematical Logic Quarterly*, 51:62–72, 2005.
- [46] Jack H. Lutz. A divergence formula for randomness and dimension. *Theoretical Computer Science*, 412:166–177, 2011.
- [47] Jack H. Lutz and Elvira Mayordomo. Dimensions of points in self-similar fractals. *SIAM Journal on Computing*, 38(3):1080–1112, 2008.
- [48] Jack H. Lutz and Klaus Weihrauch. Connectivity properties of dimension level sets. *Mathematical Logic Quarterly*, 54(5):483–491, 2008.
- [49] Benoît Mandelbrot. *Les objets fractals: Forme, hasard et dimension*. Flammarion, 1975.

- [50] Benoît Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman and Company, 1982.
- [51] Per Martin-Löf. The definition of random sequences. *Information and Control*, 9:602–619, 1966.
- [52] Elvira Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Information Processing Letters*, 84(1):1–3, 2002.
- [53] Philippe Moser. A zero-one subexp-dimension law for bpp. *Information Processing Letters*, 111:429–432, 2011.
- [54] Andre Nies. *Computability and Randomness*. Oxford University Press, reprint edition, 2012.
- [55] Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, first edition, 2014.
- [56] Robert Rettinger and Xizhong Zheng. Points on computable curves of computable lengths. In *MFCS*, pages 736–743. Springer, 2009.
- [57] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, 1987.
- [58] Claus-Peter Schnorr. A unified approach to the definition of random sequences. *Mathematical Systems Theory*, 5(3):246–258, 1971.
- [59] Claus-Peter Schnorr. *Zufälligkeit und Wahrscheinlichkeit: Eine algorithmische Begründung der Wahrscheinlichkeitstheorie*. Springer-Verlag, 1971 edition, 1971.
- [60] Claus-Peter Schnorr. Process complexity and effective random tests. *Journal of Computer and System Science*, 7:376–388, 1973.

- [61] Claus-Peter Schnorr. A survey of the theory of random sequences. In *Proceedings of the Fifth International Congress of Logic, Methodology and Philosophy of Science*, pages 193–211. Springer, 1977.
- [62] Alexander Shen and Nikolai K. Vereshchagin. Logical operations and Kolmogorov complexity. *Theoretical Computer Science*, 271(1-2):125–129, 2002.
- [63] Robert I. Soare. *Recursively Enumerable Sets and Degrees: A Study of Computable Functions and Computably Generated Sets*. Springer-Verlag, 1987.
- [64] Robert I. Soare. Turing oracle machines, online computing, and three displacements in computability theory. *Annals of Pure and Applied Logic*, 160:368–399, 2009.
- [65] R.J. Solomonoff. A preliminary report on a general theory of inductive inference. Technical Report V-131, Zator Company, Cambridge, MA, February 1960.
- [66] R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7:1–22, 1964.
- [67] R.J. Solomonoff. A formal theory of inductive inference part ii. *Information and Control*, 7:224–254, 1964.
- [68] Daniel Turetsky. Connectedness properties of dimension level sets. *Theoretical Computer Science*, 412(29):3598–3603, 2011.
- [69] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. A correction. *Proceedings of the London Mathematical Society*, 43(2):544–546, 1937.
- [70] M. van Lambalgen. *Random Sequences*. PhD thesis, University of Amsterdam, 1987.
- [71] M. van Lambalgen. Von mises’ definition of random sequences reconsidered. *Journal of Symbolic Logic*, 52(3):725–755, 1987.

- [72] V.G. Vovk. On a criterion for randomness. *Dokl. Akad. Nauk SSSR*, 294(6):1298–1302, 1987.
- [73] Klaus Weihrauch. *Computable Analysis: An Introduction*. Springer, first edition, 2000.