Dimension Extractors

David Doty[*]
Department of Computer Science
Iowa State University
Ames, IA 50011, USA
ddoty *at* iastate *dot* edu

Abstract

A *dimension extractor* is an algorithm designed to increase the effective dimension – i.e., the computational information density – of an infinite sequence. A constructive dimension extractor is exhibited by showing that every sequence of positive constructive dimension is Turing equivalent to a sequence of constructive strong dimension arbitrarily close to 1. Similar results are shown for computable dimension and truth-table equivalence, and for $\mathrm{p}_i$space dimension and $\mathrm{p}_i$space Turing equivalence, where $\mathrm{p}_i$space represents Lutz's hierarchy of super-polynomial space bounds. Thus, with respect to constructive, computable, and $\mathrm{p}_i$space information density, any sequence in which *almost every* prefix has information density *bounded away from zero* can be used to compute a sequence in which *infinitely many* prefixes have information density that is *nearly maximal.* In the constructive dimension case, the reduction is uniform with respect to the input sequence: a single oracle Turing machine, taking as input a rational upper bound on the dimension of the input sequence, works for every input sequence of positive constructive dimension.

As an application, the resource-bounded extractors are used to characterize the computable dimension of individual sequences in terms of compression via truth-table reductions and to characterize the $\mathrm{p}_i$space dimension of individual sequences in terms of compression via $\mathrm{p}_i$space-bounded Turing reductions, in analogy to previous known results connecting effective dimensions to compression with effective reductions.

[*]This research was funded in part by grant number 9972653 from the National Science Foundation as part of their Integrative Graduate Education and Research Traineeship (IGERT) program.

1 Introduction

An *extractor* is an algorithm used to transform a source of weak randomness into a source of stronger randomness. Extractors are motivated in part by the abundance of weak random sources in nature – for instance, electrical noise from Zener diodes – and the need for uniform (i.e., strong) random sources in probabilistic algorithms. Von Neumann's [von63] coin flip technique is the simplest and most famous extractor: a biased coin may be used to simulate an unbiased coin by always flipping the coin twice, ignoring the combinations HH and TT, and interpreting HT to mean H and TH to mean T.

In computational complexity, a extractor is a function, computable in some resource bound such as polynomial time, taking as input a string drawn from a probability distribution X on $\{0,1\}^n$ with *min-entropy* at least k, and a much smaller string, called the *seed*, drawn from a truly uniform distribution. The extractor's output is "close" to uniformly distributed, but much larger than the seed. The min-entropy of X is defined $\min_{x \in \{0,1\}^n} \log \Pr[x = X]^{-1}$; it is the Shannon self-information [Sha48] of the string with the highest probability in $\{0,1\}^n$. If k is strictly between 0 and n, X may be thought of as "partially random"; n bits drawn from X have k bits of randomness. The goal of an extractor is to transform X into a distribution that is closer to "fully random", i.e., to output m bits that have close to m bits of randomness. See [Sha02] for a survey of extractors in computational complexity.

For algorithmic purposes, a deterministic infinite sequence that *appears* random to any algorithm often works just as well as a truly random source. The complexity class BPP, defined by Gill [Gil77] to be those languages decidable by a randomized polynomial time algorithm with probability of correctness at least 2/3, is generally regarded as the set of decision problems feasibly decidable by a randomized algorithm. Bennett [Ben88] has demonstrated that, given access to *any* oracle sequence that is algorithmically random in the sense of Martin-Löf [Mar66], every language in BPP can be decided *deterministically* in polynomial time. Book, Lutz and Wagner [BLW94] have shown a wealth of similar characterizations of BPP and other randomized complexity classes in terms of oracle access to Martin-Löf random sequences. Lutz [Lut93], using the techniques of resource-bounded measure [Lut92], improved the result of Bennett by showing that all of BPP can be decided in polynomial time relative to any *pspace-random* oracle, which is a sequence that appears random to any polynomial-space-bounded algorithm.

Lutz refined the theory of resource-bounded measure by introducing *effective dimension* [Lut03a, Lut03b], an effectivization of classical Hausdorff dimension [Hau19, Fal03]; Athreya, Hitchcock, Lutz, and Mayordomo [AHLM] defined *effective strong dimension*, an effectivization of packing dimension [Tri82, Sul84, Fal03], showing it to be an exact dual of Hausdorff dimension. Given a resource bound Δ (such as computable, polynomial time, polynomial space, etc.), any sequence that is Δ-random has Δ-dimension 1, whereas a Δ-non-random sequence may have Δ-dimension anywhere in the interval [0,1], quantifying how close the sequence is to being Δ-random. Gu and Lutz [GL05] improved Lutz's above-mentioned pspace-random oracle result by showing that all of BPP is polynomial time decidable relative to any oracle sequence with positive pspace-dimension. Therefore, for certain applications, if a sequence has positive dimension, it contains sufficient randomness to act as a truly random source. This highlights the parallels between the theory of effective dimension and randomness extractors. A non-random sequence with positive dimension may be considered "weakly random": the first n bits of a sequence with Hausdorff and packing Δ-dimension equal to α contain about αn bits of Δ-randomness.

A well-studied effective dimension is constructive dimension [Lut03b], which has a simple characterization shown by Mayordomo [May02]. The *(constructive) dimension* of a sequence S is $\dim(S) = \liminf_{n\to\infty} \frac{\mathrm{K}(S\upharpoonright n)}{n}$, and the *(constructive) strong dimension* of S is $\mathrm{Dim}(S) = \limsup_{n\to\infty} \frac{\mathrm{K}(S\upharpoonright n)}{n}$, where $\mathrm{K}(S \upharpoonright n)$ is the Kolmogorov complexity of the n^{th} prefix of S (see [LV97]). Any Martin-Löf random sequence has constructive dimension 1, though the converse does not hold. Reimann [Rei04] and Terwijn asked the question, given any sequence S such that $\dim(S) > 0$, does oracle access to S allow us to compute a Martin-Löf random sequence? Miller and Nies [MN05] posed the related questions, does oracle access to S allow us to compute a sequence of constructive dimension 1, or arbitrarily close to 1, or strictly greater than $\dim(S)$? Viewing dimension as a quantification of the amount of randomness contained in a sequence, a computation increasing the dimension of a sequence performs the same function as the extractors mentioned earlier: the computation transforms a partially random source into a more random source.

The questions raised by Reimann, Terwijn, Miller, and Nies remain elusive, but we show that constructive dimension can be extracted in a weaker sense. We demonstrate that, for every $\epsilon > 0$ and every sequence S such that $\dim(S) > 0$, there is a sequence P, Turing equivalent to S, such that $\mathrm{Dim}(P) \geq 1 - \epsilon$. Recalling the characterizations of dim and Dim above, this means that every sequence in which *almost every* prefix has randomness *bounded away from zero* can be used to compute a sequence that has *infinitely many* prefixes with *nearly maximal* randomness. In fact, there is a single oracle Turing machine that accomplishes this extraction, taking a rational $\beta > \dim(S)$ as an input parameter, where $\beta - \dim(S) \leq \epsilon \cdot \dim(S)/3$ guarantees that $\mathrm{Dim}(P) \geq 1 - \epsilon$. Moreover, the extractor uses close to an optimal number of bits of the input sequence to compute the output sequence.

Unfortunately, our technique probably cannot be used to show that Turing reductions are able to increase constructive dimension in addition to constructive strong dimension (i.e., to show that *almost every* prefix of the output sequence has high Kolmogorov complexity). Nies and Reimann [NR06] have shown that constructive dimension *cannot* be extracted with weak truth-table reductions: for every rational α, there is a sequence S such that $\dim(S) = \alpha$, and every sequence P that is weak truth-table reducible to S satisfies $\dim(P) \leq \alpha$. Since the Turing reduction in our proof is also a weak truth-table reduction, it is unlikely that our technique is easily modified to show that $\dim(P) \geq 1-\epsilon$.

Buhrman, Fortnow, Newman, and Vereshchagin [BFNV05] and Fortnow, Hitchcock, Pavan, Vinodchandran, and Wang [FHP$^+$06] have demonstrated related constructions for extracting Kolmogorov complexity from finite strings. Buhrman et al. show that there is an efficient algorithm, taking as input any non-random string, that outputs a small list of strings of the same length as the input string, where at least one output string is guaranteed to have higher Kolmogorov complexity than the input. Note that given $x \in \{0,1\}^*$ and $\mathrm{K}(x)$, a random string containing exactly the amount of algorithmic information in x may be extracted from x: namely, the shortest program for x. The value $\mathrm{K}(x)$ – requiring at most $\log|x|$ bits to represent – may be considered "advice" bits that help the algorithm extract randomness from x. Fortnow et al. improve upon this observation by showing that there is an efficient algorithm such that, for any $0 < \alpha < \beta < 1$, if the input string x has Kolmogorov complexity at least $\alpha|x|$, then, given a *constant* (with respect to α and β) number of advice bits, the output string y (with $|y| = \Omega(|x|)$) will have Kolmogorov complexity at least $\beta|y|$. The advice bits are a necessity; Vereshchagin and Vyugin [VV02] have shown that there is no uniform algorithm capable of extracting Kolmogorov complexity from finite strings.

We also construct extractors for Δ-dimension in a similar fashion to constructive dimension, where Δ represents either the class comp of computable functions or, for each $i \in \mathbb{N}$, any of the classes $\mathrm{p}_i$space of $\mathrm{p}_i$space-computable functions. The $\mathrm{p}_i$space hierarchy was defined by Lutz [Lut92] as a hierarchy of classes of functions computable in super-polynomial, but sub-exponential, space. For instance, $\mathrm{p}_1$space is simply pspace, the class of functions computable in space n^k for some constant k, and $\mathrm{p}_2$space is the class of functions computable in space $n^{(\log n)^k}$. For every $\epsilon > 0$ and every sequence S such that $\dim_\Delta(S) > 0$, there is a sequence P, Δ-Turing equivalent to S, such that $\mathrm{Dim}_\Delta(P) \geq 1 - \epsilon$. (Precise definitions of the Δ bounds follow in section 2.3.) The only difference from the constructive dimension result is that a different oracle Turing machine is required for each sequence S. For both the constructive dimension and Δ-dimension extractors, the intuition behind the proof is the same. The extractor acts as a compressor that compresses the input sequence close to its optimal compression ratio under the resource bound Δ, which is precisely the Δ-dimension of the sequence. It is well-known [LV97] that the shortest program to produce a finite string must itself be incompressible (i.e., have maximal Kolmogorov complexity). Similarly, Mayordomo's Kolmogorov complexity characterization of constructive dimension and Hitchcock's Δ-bounded Kolmogorov complexity characterization of Δ-dimension [Hit03] are invoked to show that an almost optimally compressed representation of a sequence must itself be infinitely often nearly incompressible and thus have strong dimension close to 1.

As an application of the extractors, we use the Δ-dimension extractors to show a new characterization of Δ-dimension: the Δ-dimension of a sequence is the optimal compression ratio achievable on the sequence with a Δ-bounded Turing reduction. In the case of $\Delta =$ comp, a Δ-bounded Turing reduction is exactly a truth-table reduction [Soa87]. Thus, the computable dimension of a sequence is the optimal compression ratio achievable on it with truth-table reductions, and the $\mathrm{p}_i$space dimension of a sequence is the optimal compression ratio achievable on it with Turing reductions computable in $\mathrm{p}_i$space.

In each case, the extractor is no more powerful than the resource bound defining the dimension. This is necessary to make the results non-trivial. For instance, without access to any oracle sequence, but given exponential space, a program can diagonalize against all pspace-bounded martingales to compute a pspace-random sequence. An extractor is interesting only when it has no more computational power than the class of algorithms it is trying to fool: all the randomness present in the output sequence must originate from the input sequence, and the extractor merely acts as a filter that distills the randomness out from the redundancy.

2 Preliminaries

We refer the reader to [LV97] for an introduction to Kolmogorov complexity and algorithmic information theory, [Soa87] for an introduction to computability theory, and [Pap94] for an introduction to computational complexity theory.

2.1 Notation

All logarithms are base 2. We write $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Q}_2$, $\mathbb{Z}$, and $\mathbb{N}$ for the set of all reals, rationals, dyadic rationals, integers, and non-negative integers, respectively. For all $A \subseteq \mathbb{R}$, A^+ denotes $A \cap (0, \infty)$. $\{0,1\}^*$ denotes the set of all finite, binary *strings*. For all $x \in \{0,1\}^*$, $|x|$ denotes the *length* of x. λ denotes the empty string. Let $\sigma_0, \sigma_1, \sigma_2, \ldots \in \{0,1\}^*$ denote the standard enumeration of binary strings $\sigma_0 = \lambda, \sigma_1 = 0, \sigma_2 = 1, \sigma_3 = 00, \ldots$. For $k \in \mathbb{N}$, $\{0,1\}^k$ denotes the set of all strings $x \in \{0,1\}^*$ such that $|x| = k$. $\mathbf{C} = \{0,1\}^\infty$ denotes the *Cantor space*, the set of all infinite, binary *sequences*. For $x \in \{0,1\}^*$ and $y \in \{0,1\}^* \cup \mathbf{C}$, xy denotes the concatenation of x and y, $x \sqsubseteq y$ denotes that x is a *prefix* of y; i.e., there exists $u \in \{0,1\}^* \cup \mathbf{C}$ such that $xu = y$, and $x \sqsubset y$ denotes that $x \sqsubseteq y$ and $x \neq y$. For $S \in \{0,1\}^* \cup \mathbf{C}$ and $i, j \in \mathbb{N}$, $S[i]$ denotes the i^{th} bit of S, with $S[0]$ being the leftmost bit, $S[i \,..\, j]$ denotes the substring consisting of the i^{th} through j^{th} bits of S (inclusive), with $S[i \,..\, j] = \lambda$ if $i > j$, and $S \upharpoonright i$ denotes $S[0 \,..\, i-1]$. A *language* is a subset of $\{0,1\}^*$, and we identify a language $L \subseteq \{0,1\}^*$ with its *characteristic sequence* $\chi_L \in \mathbf{C}$, where the n^{th} bit of χ_L is 1 if and only if $\sigma_n \in L$, writing $L \upharpoonright i$ to denote $\chi_L \upharpoonright i$.

2.2 Kolmogorov Complexity and Coding

Fix a self-delimiting universal Turing machine U. Let $w \in \{0,1\}^*$. The *Kolmogorov complexity* of w is

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

The quantity $\frac{\mathrm{K}(w)}{|w|}$ is called the *Kolmogorov rate* of w. Given a bound $t : \mathbb{N} \to \mathbb{N}$, the *t-time-bounded Kolmogorov complexity* of w is

$$\mathrm{K}^t(w) = \min_{\pi \in \{0,1\}^*} \{\, |\pi| \mid U(\pi) = w \text{ in at most } t(|w|) \text{ time} \,\},$$

and the *t-space-bounded Kolmogorov complexity* of w is

$$\mathrm{KS}^t(w) = \min_{\pi \in \{0,1\}^*} \{\, |\pi| \mid U(\pi) = w \text{ in at most } t(|w|) \text{ space} \,\}.$$

Fact 2.1. *For all $w \in \{0,1\}^*$ and $t : \mathbb{N} \to \mathbb{N}$, $\mathrm{K}(w) \leq \mathrm{KS}^t(w) \leq \mathrm{K}^t(w)$.*

For all $q \in \mathbb{Q}$, let $\mathrm{K}(q) = \mathrm{K}(b(q))$, $\mathrm{K}^t(q) = \mathrm{K}^t(b(q))$ and $\mathrm{KS}^t(q) = \mathrm{KS}^t(b(q))$, where $b(q) \in \{0,1\}^*$ is some standard binary representation of the rational q with a numerator, denominator, and sign bit.

For all $w \in \{0,1\}^*$, let $e_0(w) = 0^{|w|}1w$. Define the *self-delimiting encoding function* $\mathrm{enc} : \{0,1\}^* \to \{0,1\}^*$ for all $w \in \{0,1\}^*$ by

$$\mathrm{enc}(w) = e_0\left(\sigma_{|w|}\right) w.$$

For all $n \in \mathbb{N}$, let $\mathrm{enc}(n) = \mathrm{enc}(\sigma_n)$. Strings encoded by enc and valid programs for U are *self-delimiting.* They can be prepended to arbitrary strings and uniquely decoded.

Observation 2.2. *For all $w, x \in \{0,1\}^*$, $|\mathrm{enc}(w)| \le |w| + 2\log|w| + 3$, and for all $n \in \mathbb{N}$, $\mathrm{enc}(n) \le \log n + 2\log\log n + 3$.*

2.3 Space Bounds, Reductions, and Compression

The following explanation of growth rates and function classes is taken nearly verbatim from [Lut92].

For each $i \in \mathbb{N}$, define a class G_i of *growth rates* as follows.

$$\begin{aligned} G_0 &= \{\, f : \mathbb{N} \to \mathbb{N} \mid (\exists k \in \mathbb{N})(\forall^\infty n \in \mathbb{N})\ f(n) \le kn \,\} \\ G_{i+1} &= 2^{G_i(\log n)} = \left\{\, f : \mathbb{N} \to \mathbb{N} \;\middle|\; (\exists g \in G_i)(\forall^\infty n \in \mathbb{N})\ f(n) \le 2^{g(\log n)} \,\right\} \end{aligned}$$

In this paper, for each $i \in \mathbb{N}$, Δ represents any of the following classes of functions

$$\begin{aligned} \mathrm{comp} &= \{\, f : \{0,1\}^* \to \{0,1\}^* \mid f \text{ is computable } \}, \\ \mathrm{p}_i\mathrm{space} &= \{\, f : \{0,1\}^* \to \{0,1\}^* \mid f \text{ is computable in } G_i \text{ space } \}. \end{aligned}$$

For example, $\mathrm{p}_0$space is the set of all functions computable in linear space, and $\mathrm{p}_1$space, abbreviated pspace, is the set of all functions computable in polynomial space. Given a class of functions Δ, let $\mathrm{sb}(\Delta) \subseteq \{f : \mathbb{N} \to \mathbb{N}\}$ denote the class of space bounds for Δ: sb(comp) is the set of all computable functions $f : \mathbb{N} \to \mathbb{N}$, and, for all $i \in \mathbb{N}$, $\mathrm{sb}(\mathrm{p}_i\mathrm{space}) = G_i$.

If D is a discrete domain, we say a function $f : D \to \mathbb{R}$ is *Δ-computable* if there is a function $\widehat{f} : \mathbb{N} \times D \to \mathbb{Q}$ such that $|\widehat{f}(r,x) - f(x)| \le 2^{-r}$ for all $r \in \mathbb{N}$ and $x \in D$, and $\widehat{f} \in \Delta$ (with r coded in unary and x and the output coded in binary). We say that f is *exactly Δ-computable* if $f : D \to \mathbb{Q}$ and $f \in \Delta$, and we say that f is *dyadically Δ-computable* if $f : D \to \mathbb{Q}_2$ and $f \in \Delta$.

Let M be a Turing machine and $S \in \mathbf{C}$. We say M *computes* S if, on input $n \in \mathbb{N}$, M outputs the string $S \upharpoonright n$. We define an *oracle Turing machine* (OTM) to be a Turing machine M that can make constant-time queries to an oracle sequence, and we let OTM denote the set of all oracle Turing machines. For $R \in \mathbf{C}$, we say M operates *with oracle* R if, whenever M makes a query to index $n \in \mathbb{N}$, the bit $R[n]$ is returned. We write M^R to denote the OTM M with oracle R.

Let $S, R \in \mathbf{C}$ and $M \in$ OTM. We say *S is Turing reducible to R via M*, and we write *$S \le_{\mathrm{T}} R$ via M*, if M^R computes S. We say *S is Turing reducible to R*, and we write $S \le_{\mathrm{T}} R$, if there exists $M \in$ OTM such that $S \le_{\mathrm{T}} R$ via M. We say *S is Δ-Turing reducible to R via M*, and we write *$S \le_{\mathrm{T}}^{\Delta} R$ via M*, if M^R computes S, and there is a function $q \in \mathrm{sb}(\Delta)$ such that, for all $n \in \mathbb{N}$, M outputs $S \upharpoonright n$ using at most $q(n)$ space. We say *S is Δ-Turing reducible to R*, and we write $S \le_{\mathrm{T}}^{\Delta} R$, if there exists $M \in$ OTM such that $S \le_{\mathrm{T}}^{\Delta} R$ via M. We say S is *Turing equivalent* to R, and we write $S \equiv_{\mathrm{T}} R$, if $S \le_{\mathrm{T}} R$ and $R \le_{\mathrm{T}} S$, and we say S is *Δ-Turing equivalent* to R, and we write $S \equiv_{\mathrm{T}}^{\Delta} R$, if $S \le_{\mathrm{T}}^{\Delta} R$ and $R \le_{\mathrm{T}}^{\Delta} S$.

If $\Delta = \text{comp}$, then a Δ-Turing reduction is nothing more than a *truth-table reduction* (see [Soa87]). We write $S \leq_{\text{tt}} R$ to denote that S is truth-table reducible to R (i.e., that $S \leq_{\text{T}}^{\text{comp}} R$). If $\Delta = \text{p}_i\text{space}$, and we identify a sequence $S \in \mathbf{C}$ with the language $L \subseteq \{0,1\}^*$ for which $S = \chi_L$, then a Δ-Turing reduction is an E_iSPACE (see [Lut92]) Turing reduction. Since this paper deals exclusively with sequences, we will use the convention of calling such a reduction a $\text{p}_i$space-Turing reduction, indicating that the polynomial bound is in terms of the length of the prefix of the characteristic sequence (the output) and not in terms of the length of the strings in the language (the input).

Let $S, R \in \mathbf{C}$ and $M \in \text{OTM}$ such that $S \leq_{\text{T}} R$ via M. Define $\#(M^R, S \upharpoonright n)$ to be the *query usage of M^R on $S \upharpoonright n$*, the number of bits of R queried by M when computing the string $S \upharpoonright n$. (If we instead define $\#(M^R, S \upharpoonright n)$ to be the index of the rightmost bit of R queried by M when computing $S \upharpoonright n$, all results of the present paper still hold.) Define

$$\begin{aligned}\rho_M^-(S,R) &= \liminf_{n\to\infty} \frac{\#(M^R, S \upharpoonright n)}{n},\\ \rho_M^+(S,R) &= \limsup_{n\to\infty} \frac{\#(M^R, S \upharpoonright n)}{n}.\end{aligned}$$

Viewing R as a compressed version of S, $\rho_M^-(S,R)$ and $\rho_M^+(S,R)$ are respectively the best- and worst-case compression ratios as M decompresses R into S. Note that $0 \leq \rho_M^-(S,R) \leq \rho_M^+(S,R) \leq \infty$. For $S \in \mathbf{C}$, the *lower and upper Turing compression ratios of S* are respectively defined

$$\begin{aligned}\rho^-(S) &= \min_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^-(S,R) \;\middle|\; S \leq_{\text{T}} R \text{ via } M \right\},\\ \rho^+(S) &= \min_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^+(S,R) \;\middle|\; S \leq_{\text{T}} R \text{ via } M \right\}.\end{aligned}$$

It was shown in [Dot06] that the above minima exist. Note that $0 \leq \rho_{\text{T}}^-(S) \leq \rho_{\text{T}}^+(S) \leq 1$. The *lower and upper Δ-Turing compression ratios of S* are respectively defined

$$\begin{aligned}\rho_\Delta^-(S) &= \inf_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^-(S,R) \;\middle|\; S \leq_{\text{T}}^{\Delta} R \text{ via } M \right\},\\ \rho_\Delta^-(S) &= \inf_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^+(S,R) \;\middle|\; S \leq_{\text{T}}^{\Delta} R \text{ via } M \right\}.\end{aligned}$$

Recall that a $\leq_{\text{T}}^{\text{comp}}$-reduction is simply a truth-table reduction. Therefore, for all $S \in \mathbf{C}$, the *lower and upper truth-table compression ratios* of S are respectively defined

$$\begin{aligned}\rho_{\text{tt}}^-(S) &= \rho_{\text{comp}}^-(S) = \inf_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^-(S,R) \;\middle|\; S \leq_{\text{tt}} R \text{ via } M \right\},\\ \rho_{\text{tt}}^+(S) &= \rho_{\text{comp}}^+(S) = \inf_{\substack{R\in\mathbf{C}\\ M\in\text{OTM}}} \left\{ \rho_M^+(S,R) \;\middle|\; S \leq_{\text{tt}} R \text{ via } M \right\}.\end{aligned}$$

2.4 Effective Dimension

See [Lut03a, Lut03b] for an introduction to the theory of effective dimension.

We use Mayordomo's characterization [May02] of the constructive dimensions of sequences. For all $S \in \mathbf{C}$, the *(constructive) dimension* and the *(constructive) strong dimension* of S are respectively defined

$$\dim(S) = \liminf_{n\to\infty} \frac{\mathrm{K}(S \upharpoonright n)}{n}, \text{ and } \mathrm{Dim}(S) = \limsup_{n\to\infty} \frac{\mathrm{K}(S \upharpoonright n)}{n}.$$

Other equivalent definitions exist, in terms of martingales [Lut03b] and Turing reduction compression [Dot06], but we use only Mayordomo's characterization in the present paper.

Resource-bounded dimension was first defined in [Lut03a]. It is based on martingales, which are strategies for betting on bits of an infinite sequence.

1. An *s-gale* is a function $d : \{0,1\}^* \to [0,\infty)$ such that, for all $w \in \{0,1\}^*$,
$$d(w) = 2^{-s}[d(w0) + d(w1)].$$
2. A *martingale* is a 1-gale.

Intuitively, a martingale is a strategy for gambling in the following game. The gambler starts with some initial amount of *capital* (money) $d(\lambda)$, and it reads an infinite sequence S of bits. $d(w)$ represents the capital the gambler has after reading the prefix $w \sqsubset S$. Based on w, the gambler bets some fraction of its capital that the next bit will be 0 and the remainder of its capital that the next bit will be 1. The capital bet on the bit that appears next is doubled, and the remaining capital is lost. The condition $d(w) = \frac{d(w0)+d(w1)}{2}$ ensures *fairness*: the martingale's expected capital after seeing the next bit, given that it has already seen the string w, is equal to its current capital. The fairness condition and an easy induction lead to the following observation.

Observation 2.3. *Let $k \in \mathbb{N}$ and let $d : \{0,1\}^* \to [0,\infty)$ be a martingale. Then*

$$\sum_{u\in\{0,1\}^k} d(u) = 2^k d(\lambda).$$

An s-gale is a martingale in which the capital bet on the bit that occurred is multiplied by 2^s, as opposed to simply 2, after each bit. The parameter s may be regarded as the *unfairness of the betting environment*; the lower the value of s, the faster money is taken away from the gambler. Let $d : \{0,1\}^* \to [0,\infty)$ be a martingale and let $s \in [0,\infty)$. Define the *s-gale induced by d*, denoted $d^{(s)}$, for all $w \in \{0,1\}^*$ by

$$d^{(s)}(w) = 2^{(s-1)|w|} d(w).$$

If a gambler's martingale is given by d, then, for all $s \in [0,\infty)$, its s-gale is $d^{(s)}$.

The following theorem, due to Lutz, establishes an upper bound on the number of strings on which an s-gale can perform well.

Theorem 2.4. [Lut03a] *Let d be an s-gale. Then for all $w \in \{0,1\}^*$, $k \in \mathbb{N}$, and $\alpha \in \mathbb{R}^+$, there are fewer than $\frac{2^k}{\alpha}$ strings $u \in \{0,1\}^k$ for which*

$$\max_{v\sqsubseteq u} \left\{2^{(1-s)|v|} d(wv)\right\} \geq \alpha d(w).$$

Corollary 2.5. *Let d be a martingale. Then for all $l \in \mathbb{R}$, $w \in \{0,1\}^*$, $k \in \mathbb{N}$, and $\alpha \in \mathbb{R}^+$, there are fewer than $\frac{2^l}{\alpha}$ strings $u \in \{0,1\}^k$ for which*

$$d(wu) \geq \alpha 2^{k-l} d(w).$$

Let $S \in \mathbf{C}$, $s \in [0, \infty)$, and let $d : \{0,1\}^* \to [0, \infty)$ be an s-gale. d *succeeds* on S, and we write $S \in \mathrm{S}^\infty[d]$, if

$$\limsup_{n\to\infty} d(S \upharpoonright n) = \infty.$$

d *strongly succeeds* on S, and we write $S \in \mathrm{S}^\infty_{\mathrm{str}}[d]$, if

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

The following lemma follows easily from the proof of the Exact Computation Lemma of [Lut03a].

Lemma 2.6. [Lut03a] *If d is a Δ-computable s-gale and 2^s is a dyadic rational, then there is a dyadically Δ-computable s-gale $\widetilde{d}$ such that $\mathrm{S}^\infty[d] \subseteq \mathrm{S}^\infty[\widetilde{d}]$ and $\mathrm{S}^\infty_{\mathrm{str}}[d] \subseteq \mathrm{S}^\infty_{\mathrm{str}}[\widetilde{d}]$.*

Let $G^{(s)}_\Delta$ denote the set of all Δ-computable s-gales. For all $S \in \mathbf{C}$, the Δ-*dimension* and the Δ-*strong dimension* of S are respectively defined

$$\dim_\Delta(S) = \inf\left\{ s \in [0,\infty) \;\middle|\; \left(\exists d \in G^{(s)}_\Delta\right)\; S \in \mathrm{S}^\infty[d] \right\},$$

and

$$\mathrm{Dim}_\Delta(S) = \inf\left\{ s \in [0,\infty) \;\middle|\; \left(\exists d \in G^{(s)}_\Delta\right)\; S \in \mathrm{S}^\infty_{\mathrm{str}}[d] \right\}.$$

The following alternate characterization of the Δ-dimensions is due to Hitchcock [Hit03].

Theorem 2.7. [Hit03] *For all $S \in \mathbf{C}$,*

$$\dim_\Delta(S) = \inf_{p \in \mathrm{sb}(\Delta)} \liminf_{n\to\infty} \frac{\mathrm{KS}^p(S \upharpoonright n)}{n},$$

and

$$\mathrm{Dim}_\Delta(S) = \inf_{p \in \mathrm{sb}(\Delta)} \limsup_{n\to\infty} \frac{\mathrm{KS}^p(S \upharpoonright n)}{n}.$$

If there is a martingale d that succeeds on a sequence $S \in \mathbf{C}$, then d makes arbitrarily high capital on S. Using a standard technique (see [MM04]), one may construct from d a martingale d' that *strongly* succeeds on S. This is done by maintaining a "side account" of capital that is not used to bet: i.e., the capital in that account is always allocated equally between 0 and 1 when betting. Whenever d makes strictly more than \$1, \$1 is moved into the side account. Since d succeeds on S, it will eventually make more than \$1 in the main account again, and so infinitely often, the side account will grow by \$1, whence d' strongly succeeds on S. It is clear that if d is Δ-computable, then d' is also Δ-computable.

Observation 2.8. *Let $S \in \mathbf{C}$ such that there is a Δ-computable martingale that succeeds on S. Then there is a Δ-computable martingale that strongly succeeds on S.*

Let $r, t \in [0, \infty)$ and $S \in \mathbf{C}$. Note that if d_1 and d_2 are martingales such that $d_1^{(t)}$ succeeds on S and $d_2^{(r)}$ strongly succeeds on S, then d_1 and d_2 can be combined into a single martingale d that simulates d_1 and d_2 in separate "accounts". Furthermore, if d_1 and d_2 are both Δ-computable, then d is also Δ-computable. This leads to the following observation.

Observation 2.9. *Let $S \in \mathbf{C}$, $t > \dim_\Delta(S)$, and $r > \mathrm{Dim}_\Delta(S)$. Then there is a Δ-computable martingale d such that $d^{(t)}$ succeeds on S and $d^{(r)}$ strongly succeeds on S.*

The next theorem is due to Ryabko, though it has been re-phrased from his original formulation.

Theorem 2.10. [Rya84, Rya86] *There exist OTMs M_e and M_d, with M_e taking a single input $\beta \in \mathbb{Q}$, with the property that, for every $S \in \mathbf{C}$ and every rational $\beta > \dim(S)$, there exists $P \in \mathbf{C}$ such that $P \leq_\mathrm{T} S$ via $M_e(\beta)$, $S \leq_\mathrm{T} P$ via M_d, and $\rho^-_{M_d}(S, P) < \beta$.*

3 Dimension Extractors

This section explores three effective dimensions in which effective reducibilities may be used to extract effective strong dimension: constructive dimension, extracted with Turing reductions, computable dimension, extracted with truth-table reductions, and p$_i$space-dimension, extracted with p$_i$space-bounded Turing reductions.

3.1 Constructive Dimension Extractors

The next theorem states that any sequence in which almost every prefix has Kolmogorov rate bounded away from zero can be used to compute a sequence with infinitely many prefixes of nearly maximal Kolmogorov rate. Furthermore, this can be done with a single OTM taking a rational upper bound on the dimension of the input sequence.

Theorem 3.1. *There exist $N_e, N_d \in$ OTM, with N_e taking $\beta \in \mathbb{Q}$ as input, such that, for all $S \in \mathbf{C}$ such that $\dim(S) > 0$, and all $\epsilon > 0$ such that $0 < \beta - \dim(S) \leq \epsilon \cdot \dim(S)/3$, there exists $P \in \mathbf{C}$ such that $P \leq_\mathrm{T} S$ via $N_e(\beta)$, $S \leq_\mathrm{T} P$ via N_d, and $\mathrm{Dim}(P) \geq 1 - \epsilon$.*

The statement of Theorem 3.1 is complicated by the rational input β required to make the OTM N_e uniform over all sequences. The following corollary states simply that Turing reductions can extract strong dimension from positive dimension.

Corollary 3.2. *For each $S \in \mathbf{C}$ such that $\dim(S) > 0$, and each $\epsilon > 0$, there exists $P \in \mathbf{C}$ such that $P \equiv_\mathrm{T} S$ and $\mathrm{Dim}(P) \geq 1 - \epsilon$.*

Proof of Theorem 3.1. Let S, β, and ϵ be as in the statement of the theorem, and define $\delta = \beta - \dim(S) > 0$. Let $M_e, M_d \in \mathrm{OTM}$ be as in Theorem 2.10, so that $M_e^S(\beta)$ computes P, M_d^P computes S, and, letting $p_i = \#(M_d^P, S \upharpoonright i)$ for all $i \in \mathbb{N}$,

$$\begin{aligned} \liminf_{i\to\infty} \frac{p_i}{i} < \beta \quad &\Longrightarrow \quad (\exists^\infty i \in \mathbb{N}) \ \liminf_{n\to\infty} \frac{\mathrm{K}(S \upharpoonright n)}{n} > \frac{p_i}{i} - \delta \\ &\Longrightarrow \quad (\exists^\infty n \in \mathbb{N}) \ \mathrm{K}(S \upharpoonright n) > p_n - \delta n. \end{aligned} \tag{3.1}$$

Since $\delta \leq \epsilon \cdot \dim(S)/3 < \epsilon \cdot \dim(S)/2$,

$$(\forall^\infty n \in \mathbb{N}) \ \mathrm{K}(S \upharpoonright n) > \frac{2}{\epsilon}\delta n. \tag{3.2}$$

Ryabko's construction of M_d is such that entire prefixes of the oracle sequence are queried at once: whenever the bit at index $i \in \mathbb{N}$ is queried, all bits $j < i$ are also queried. Thus, a program M simulating M_d with the first p_n bits of P can calculate $S \upharpoonright n$, and M can be encoded in $|\mathrm{enc}(P \upharpoonright p_n)| + O(1)$ bits. Thus there is a constant c such that $\mathrm{K}(S \upharpoonright n) \leq p_n + 2\log p_n + c$, which together with (3.2) implies that

$$(\forall^\infty n \in \mathbb{N}) \ \delta n < \frac{\epsilon}{2}(p_n + 2\log p_n + c). \tag{3.3}$$

Combining (3.1) and (3.3),

$$(\exists^\infty n \in \mathbb{N}) \ \mathrm{K}(S \upharpoonright n) > p_n - \frac{\epsilon}{2}\left(p_n + 2\log p_n + c\right). \tag{3.4}$$

If the OTMs $N_e(\beta)$ and N_d simulate $M_e(\beta)$ and M_d, respectively, then $N_e(\beta)$ and N_d testify that $P \equiv_{\mathrm{T}} S$. It remains to show that $\mathrm{Dim}(P) \geq 1 - \epsilon$. Suppose for the sake of contradiction that $\mathrm{Dim}(P) < 1 - \epsilon$. Then it would be the case that

$$(\forall^\infty m \in \mathbb{N}) \ \mathrm{K}(P \upharpoonright m) < m - \epsilon m. \tag{3.5}$$

Since $\dim(S) > 0$, S is uncomputable, and therefore p_n grows unboundedly with n. A program that produces $P \upharpoonright p_n$ can be used in conjunction with M_d to produce $S \upharpoonright n$. Therefore, for a suitable constant $c' \approx |M_d|$,

$$\begin{aligned} (\forall^\infty n \in \mathbb{N}) \ \mathrm{K}(S \upharpoonright n) \ &\leq \ \mathrm{K}(P \upharpoonright p_n) + c' \\ &< \ p_n - \epsilon p_n + c' \\ &< \ p_n - \frac{\epsilon}{2}\left(p_n + 2\log p_n + c\right). \end{aligned} \tag{3.6}$$

But (3.6) contradicts (3.4). Hence, $\mathrm{Dim}(P) \geq 1 - \epsilon$. □

3.2 Resource-bounded Dimension Extractors

We show that an analog of Corollary 3.2 holds for Δ-dimension. Many of the techniques in this section are resource-bounded analogs of the techniques of [Dot06], although each the techniques of that paper require at least minor adaptation for use in the current paper.

The following technical lemma is needed later in the section to facilitate the economical computation and storage of rational approximations of the values of a martingale.

Lemma 3.3. *Let $a \in \mathbb{R}^+$ be fixed, and, for all $i \in \mathbb{N}$, let $r_i \in [1, a2^{i^2}]$. Then for each $i \in \mathbb{N}$, there exists $c_i \in \mathbb{Q}_2^+$ such that $r_i\left(1 - \frac{1}{i^2}\right) \leq c_i \leq r_i$ and $\mathrm{K}^{O(i^2)}(c_i) = O(\log i)$. Furthermore, if $a, r_i \in \mathbb{Q}_2^+$, then c_i can be computed from i, a, and r_i in $O(i^2)$ time.*

Proof. We prove the cases $r_i \geq i^2$ and $1 \leq r_i < i^2$ separately. Suppose $r_i \geq i^2$. In this case we will choose c_i to be an integer. Set $k \in \mathbb{Z}^+$ such that $2^{k-1} < i^2 \leq 2^k$. Since $r_i \geq i^2 > 2^{k-1}$, $\lceil \log r_i \rceil > k - 1$.

Let $c_i \in \mathbb{Z}^+$ be the integer whose binary representation is $x0^{\lceil \log r_i \rceil - k}$, where $x \in \{0,1\}^k$ is the first k bits of $\lfloor r_i \rfloor$. Since c_i shares its first k bits with r_i,

$$r_i - c_i \leq 2^{\lceil \log r_i \rceil - k} - 1 \leq \frac{r_i + 2}{2^k} - 1 \leq \frac{r_i}{i^2},$$

so $r_i \geq c_i \geq r_i\left(1 - \frac{1}{i^2}\right)$. c_i can be fully described by the first k bits of r_i, along with the binary representation of the number $\lceil \log r_i \rceil - k$ of 0's that follow. Thus, describing c_i requires no more than $k + \log(\lceil \log r_i \rceil - k) \leq \log i^2 + 1 + \log\log a + \log i^2 = O(\log i)$ bits.

Now suppose that $1 \leq r_i < i^2$. We let c_i approximate r_i by the binary integer $\lfloor r_i \rfloor$, plus a finite prefix of the bits to the right of r_i's decimal point in binary form. If $x.S$ is the binary representation of r_i, where $x \in \{0,1\}^*$ and $S \in \mathbf{C}$, let $c_i \in \mathbb{Z}^+$ be represented by $x.y$, where $y \sqsubseteq S$.

Since $r_i < i^2$, $|x| \leq \log i^2 = O(\log i)$. We need $r_i - c_i \leq r_i/i^2$. Since $r_i - c_i \leq 2^{-|y|}$, it suffices to choose $y \sqsubseteq S$ such that $2^{-|y|} \leq r_i/i^2$, or $|y| \geq \log(i^2/r_i)$. Let $|y| = \lceil \log(i^2/r_i) \rceil = O(\log i)$, since $r_i \geq 1$. Thus $|x| + |y| = O(\log i)$, so describing c_i requires $O(\log i)$ bits.

It is easy to verify that both programs for computing c_i require $O(i^2)$ time. Let $\pi(c_i)$ be such a program. If r_i is given as a dyadic rational input, $\pi(c_i)$ is simply constructed from initial bits of r_i in either case. It follows that in the first case, $\pi(c_i)$ can be created in $O(i^2)$ time, since $\log r_i \leq i^2 + \log a$, and the first case uses only the integral part of r_i. In the second case, we have already shown that the integral part x of c_i and the fractional part y of c_i each constitute $O(\log i)$ bits of r. Therefore, in the second case, $\pi(c_i)$ can be created in $O(\log i)$ time. $\square$

An OTM that computes a sequence S, together with a finite prefix of the oracle that it queries, is a program to produce a prefix of S. Thus, the query usage of a space-bounded OTM on that prefix of S cannot be far below the space-bounded Kolmogorov complexity of the prefix of S. This is formalized in the following lemma, which bounds the compression ratio below by dimension.

Lemma 3.4. *For all $S \in \mathbf{C}$, $\rho_\Delta^-(S) \geq \dim_\Delta(S)$, and $\rho_\Delta^+(S) \geq \mathrm{Dim}_\Delta(S)$.*

Proof. Let $S, P \in \mathbf{C}$, and let $M \in \mathrm{OTM}$ such that $S \leq_{\mathrm{T}}^{\Delta} P$ via M. For $P = S$, $S \leq_{\mathrm{T}}^{\Delta} P$ via the trivial "bit-copier" OTM that always queries exactly n bits of P to compute n bits of S, so we may assume that for all but finitely many $n \in \mathbb{N}$, $\#(M^P, S \upharpoonright n) \leq n$. Thus, since M has available at least a linear amount of space, we may assume that each bit of P is queried at most once and cached, and that subsequent queries are retrieved from the cache.

Let π_M be a self-delimiting program for M, so that, for all $x \in \{0,1\}^*$, $U(\pi_M x) = M(x)$. Let $p_n \in \{0,1\}^{\#(M^P, S \upharpoonright n)}$ be the oracle bits of P queried by M on input n, in the order in which they are queried. Recall the self-delimiting encoding function enc. For each $n \in \mathbb{N}$, let $\pi_n = \pi_{M'}\pi_M \mathrm{enc}(n)\mathrm{enc}(p_n)$, where $\pi_{M'}$ is a self-delimiting program that simulates M, encoded by π_M, on input n, encoded by $\mathrm{enc}(n)$, with oracle P, encoded by $\mathrm{enc}(p_n)$. When M makes its i^{th} query to a bit of P, the bit of $p_n[i]$ is returned. Since M queries each bit of P at most once, the bit from p_n will be correct, no matter what index was queried by M, since the bits of p_n are arranged in the order in which M makes its queries.

Then $U(\pi_n) = S \upharpoonright n$, so if there exists $s \in \mathrm{sb}(\Delta)$ such that M uses at most $s(n)$ space on input n, there exists $q \in \mathrm{sb}(\Delta)$ such that, for all $n \in \mathbb{N}$, $\mathrm{KS}^q(S \upharpoonright n) \leq |\pi_n|$. By Theorem 2.7,

$$\begin{aligned}
& \dim_\Delta(S) \\
= \; & \inf_{q \in \mathrm{sb}(\Delta)} \liminf_{n\to\infty} \frac{\mathrm{KS}^q(S \upharpoonright n)}{n} \\
\leq \; & \liminf_{n\to\infty} \frac{|\pi_{M'}\pi_M \mathrm{enc}(n)\mathrm{enc}(p_n)|}{n} \\
\leq \; & \liminf_{n\to\infty} \frac{|\pi_{M'}\pi_M| + \log n + 2\log\log n + \#(M^P, S \upharpoonright n) + 2\log \#(M^P, S \upharpoonright n) + 6}{n} \\
= \; & \liminf_{n\to\infty} \frac{\#(M^P, S \upharpoonright n)}{n} \\
= \; & \rho_M^-(S, R),
\end{aligned}$$

whence $\dim(S) \leq \rho_\Delta^-(S)$. Similarly, $\mathrm{Dim}(S) \leq \rho_\Delta^+(S)$. □

For $t \in \mathbb{R}$, $c \in \mathbb{Q}$, $s \in \{0,1\}^*$, $k \in \mathbb{N}$, and $d : \{0,1\}^* \to [0,\infty)$ a t-gale, define

$$A_{d,c,s}^{(k)} = \left\{ \ u \in \{0,1\}^k \ \middle| \ d(su) \geq c \ \right\} \tag{3.7}$$

to be the set of all length-k extensions of s on which d makes at least c capital. The following lemma shows that $|A_{d,c,s}^{(k)}|$ is small on average if d makes a lot of capital on a sequence beginning with prefix s, if c is close to the capital that d has after reading k bits beyond s.

Lemma 3.5. *Let $S \in \mathbf{C}$, $r \geq t > 0$, and let d be a martingale such that $d^{(t)}$ succeeds on S and $d^{(r)}$ strongly succeeds on S. Write $S = s_0 s_1 s_2 \ldots$, where, for all $i \in \mathbb{N}$, $k_i = |s_i|$, and $n_i = |s_0 \ldots s_i|$. Let $c_i \in \mathbb{R}$ satisfy $d(S \upharpoonright n_i)g(i) \leq c_i \leq d(S \upharpoonright n_i)$, where $g(i) \in (0,1)$ satisfies $-\sum_{j=2}^{i} \log g(j) = o(n_i)$. Then*

$$\limsup_{i\to\infty} \frac{\sum_{j=0}^{i} \log \left| A_{d,c_j,S \upharpoonright n_{j-1}}^{(k_j)} \right|}{n_i} \leq r, \tag{3.8}$$

and, if $k_i = o(n_i)$, then

$$\liminf_{i\to\infty} \frac{\sum_{j=0}^{i} \log \left| A_{d,c_j,S \upharpoonright n_{j-1}}^{(k_j)} \right|}{n_i} \leq t, \tag{3.9}$$

Proof. We show that (3.9) holds, since the proof of (3.8) is similar. Let $t' > t$, and, for all $i \in \mathbb{N}$, let $A_i = A^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}$. It suffices to show that, for infinitely many $i \in \mathbb{N}$, $\sum_{j=0}^{i} \log |A_j| \leq t' n_i$. Since $d^{(t)}$ succeeds on S, for infinitely many $n \in \mathbb{N}$,

$$d(S \upharpoonright n) \geq 2^{(1-t)n} d(\lambda). \tag{3.10}$$

A martingale can at most double its capital after every bit, and each index n with $n_i \leq n < n_{i+1}$ is at most k_i bits beyond n_i. It follows that for infinitely many $i \in \mathbb{N}$,

$$d(S \upharpoonright n_i) \geq 2^{(1-t)n_i - k_i} d(\lambda). \tag{3.11}$$

For all $i \in \mathbb{N}$, set $l_i \in \mathbb{R}$ such that $d(S \upharpoonright n_i) = 2^{k_i - l_i} d(S \upharpoonright n_{i-1})$. By induction on i,

$$d(S \upharpoonright n_i) = d(\lambda) \prod_{j=0}^{i} 2^{k_j - l_j}. \tag{3.12}$$

Then, by equations (3.11) and (3.12), and the fact that $\sum_{j=0}^{i} k_i = n_i$, for infinitely many $i \in \mathbb{N}$,

$$\begin{aligned} \prod_{j=0}^{i} 2^{k_j - l_j} \geq 2^{(1-t)n_i - k_i} \quad &\Longrightarrow \quad \sum_{j=0}^{i} (k_j - l_j) \geq (1-t)n_i - k_i \\ &\Longrightarrow \quad \sum_{j=0}^{i} l_j \leq t n_i + k_i. \end{aligned}$$

Recall that $c_i \geq d(S \upharpoonright n_i) g(i) = g(i) 2^{k_i - l_i} d(S \upharpoonright n_{i-1})$. By Corollary 2.5 (take $k = k_i, l = l_i, \alpha = 1 - \frac{1}{i^2}, w = S \upharpoonright n_{i-1}$) and the definition of l_i, it follows that $|A_i| \leq 2^{l_i}/g(i)$, and so $\log |A_i| \leq l_i - \log g(i)$. Let $c_{0,1} = \log |A_0| + \log |A_1| - l_0 - l_1$. Then

$$\begin{aligned} \sum_{j=0}^{i} \log |A_j| &\leq \sum_{j=0}^{i} l_j - \sum_{j=2}^{i} \log g(i) + c_{0,1} \\ &\leq t n_i + k_i - \sum_{j=2}^{i} \log g(i) + c_{0,1} \\ &= t' n_i + (t - t') n_i + k_i - \sum_{j=2}^{i} \log g(i) + c_{0,1}. \end{aligned} \tag{3.13}$$

$t < t'$, $k_i = o(n_i)$, and $\sum_{j=2}^{i} \log g(i) = o(n_i)$, so for infinitely many i, $\sum_{j=0}^{i} \log |A_j| \leq t' n_i$.

The proof of (3.8) is similar, replacing "for infinitely many i" conditions with "for all but finitely many i." The only difference is that (3.10) holds for all but finitely many n, and so there is no need to derive (3.11). Consequently, the term k_i does not appear on the right-hand side of (3.13), and so the condition $k_i = o(n_i)$ is not necessary to show that (3.8) holds. □

Let $t \in \mathbb{R}$, $c \in \mathbb{Q}$, $s \in \{0,1\}^*$, $k \in \mathbb{N}$, and let $d : \{0,1\}^* \to [0,\infty)$ be a t-gale. If d is exactly Δ-computable and $u \in A^{(k)}_{d,c,s}$, then the following procedure computes the index of u in a lexicographical ordering of $A^{(k)}_{d,c,s}$.

```
$\mathrm{ind}^{(k)}_{d,c,s}\left(u \in \{0,1\}^k\right)$
1  $i' \leftarrow 0$
2  **for** each $u' \in \{0,1\}^k$ in lexicographical order
3        **do if** $d(su') \geq c$
4              **then if** $u' = u$
5                    **then** output $i'$ and exit
6                    **else** $i' \leftarrow i' + 1$
```

If $u \notin A^{(k)}_{d,c,s}$, $\mathrm{ind}^{(k)}_{d,c,s}(u)$ is undefined. Note that $\mathrm{ind}^{(k)}_{d,c,s}(u) \leq \left|A^{(k)}_{d,c,s}\right|$ when it is defined. The computation of $\mathrm{str}^{(k)}_{d,c,s} : \mathbb{N} \to \{0,1\}^k$, the inverse of $\mathrm{ind}^{(k)}_{d,c,s}$, is similar:

```
$\mathrm{str}^{(k)}_{d,c,s}\,(i \in \mathbb{N})$
1  $i' \leftarrow 0$
2  **for** each $u' \in \{0,1\}^k$ in lexicographical order
3        **do if** $d(su') \geq c$
4              **then if** $i' = i$
5                    **then** output $u'$ and exit
6                    **else** $i' \leftarrow i' + 1$
```

Both $\mathrm{ind}^{(k)}_{d,c,s}$ and $\mathrm{str}^{(k)}_{d,c,s}$ are *uniformly Δ-computable* for all d, c, s, and k, in the sense that each may be implemented by a single Δ-bounded Turing machine taking d, c, s, and k as auxiliary input, provided d is exactly Δ-computable.

The following theorem was shown in [Dot06].

Theorem 3.6. [Dot06] *There is an OTM M such that, for all $S \in \mathbf{C}$, there is a sequence $R \in \mathbf{C}$ such that*

1. *$S \leq_{\mathrm{T}} R$ via M.*
2. *$\rho^-_M(S,R) = \dim(S)$.*
3. *$\rho^+_M(S,R) = \mathrm{Dim}(S)$.*

The following theorem, a Δ-bounded analog of Theorem 3.6, is used to construct Δ-dimension extractors and to give a new characterization of Δ-dimension.

Theorem 3.7. *For all $S \in \mathbf{C}$ and $\delta > 0$, there is a sequence $P \in \mathbf{C}$ and an OTM M such that*

1. *$S \equiv^{\Delta}_{\mathrm{T}} P$, with $S \leq^{\Delta}_{\mathrm{T}} P$ via M.*
2. *$\rho^-_M(S,P) \leq \dim_\Delta(S) + \delta$.*
3. *$\rho^+_M(S,P) \leq \mathrm{Dim}_\Delta(S) + \delta$.*

Proof. If $\dim_\Delta(S) = 1$, then the trivial "bit-copier" OTM M suffices to compute $P = S$, where $\rho^-_M(S,P) = \rho^+_M(S,P) = \dim_\Delta(S) = \mathrm{Dim}_\Delta(S) = 1$, so assume that $\dim_\Delta(S) < 1$.

Write $S = s_0 s_1 s_2 \ldots$, where, for all $i \in \mathbb{N}$, $k_i = |s_i| = i$, and $n_i = |s_0 \ldots s_i| = \sum_{j=0}^{i} k_i = \frac{i(i+1)}{2}$. Let $t, r \in \mathbb{Q}_2$ such that $\dim_\Delta(S) < t \leq \dim_\Delta(S) + \delta$ and $\mathrm{Dim}_\Delta(S) < r \leq \mathrm{Dim}_\Delta(S) + \delta$. Then by Observation 2.9, there is a Δ-computable martingale d such that $d^{(t)}$ succeeds on S and $d^{(r)}$ strongly succeeds on S. By Observation 2.8, $\dim_\Delta(S) < 1$ implies that we may assume that d strongly succeeds on S. By Lemma 2.6, we may assume that d is dyadically Δ-computable.

By Observation 2.3, $d(S \upharpoonright n_i) \leq 2^{n_i} d(\lambda) \leq 2^{i^2} d(\lambda)$ for $i \geq 3$. Since d strongly succeeds on S, for all but finitely many i, $d(S \upharpoonright n_i) \geq 1$. For all $i \in \mathbb{N}$, choose $c_i \in \mathbb{Q}^+$ for i, a, and r as in Lemma 3.3 (taking $a = d(\lambda)$ and $r_i = d(S \upharpoonright n_i)$), and let $\pi(c_i)$ represent a program testifying that $\mathrm{KS}^{O(i^2)}(c_i) \leq O(\log i)$, which can be computed from i, $d(\lambda)$, and $d(S \upharpoonright n_i)$ in $O(i^2)$ time.

Let $P = p_0 p_1 p_2 \ldots$, where, for all $i \in \mathbb{N}$,

$$p_i = \mathrm{enc}\left(\mathrm{ind}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}(s_i)\right) \pi(c_i).$$

Because $\mathrm{str}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}$ is an inverse of $\mathrm{ind}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}$, we can write each s_i as

$$s_i = \mathrm{str}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}\left(\mathrm{ind}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}(s_i)\right).$$

Since $\mathrm{ind}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}$, $\mathrm{str}^{(k_i)}_{d,c_i,S\upharpoonright n_{i-1}}$ and d are all Δ-computable, $P \equiv^\Delta_\mathrm{T} S$. Let M be the OTM such that M^P computes S. It suffices to show that $\rho^-_M(S,P) \leq t$ and $\rho^+_M(S,P) \leq r$. Note that

$$\begin{aligned}
-\sum_{j=2}^{i} \log\left(1 - \frac{1}{j^2}\right) &= -\sum_{j=2}^{i} \log\frac{(j+1)(j-1)}{j^2} \\
&= -\sum_{j=2}^{i} \underbrace{(\log(j+1) + \log(j-1) - 2\log j)}_{\text{telescopes}} \\
&= -\log 1 + \log 2 + \log i - \log(i+1),
\end{aligned}$$

which converges as $i \to \infty$, so $g(i) = 1 - \frac{1}{i^2}$ satisfies $-\sum_{j=2}^{i} \log g(j) = o(n_i)$, whence the conditions of Lemma 3.5 are satisfied. Then

$$\begin{aligned}
\rho^+_M(S,P) &= \limsup_{n\to\infty} \frac{\#(M^P, S\upharpoonright n)}{n} \\
&= \limsup_{i\to\infty} \frac{\#(M^P, S\upharpoonright n_i)}{n_i} && \text{since } k_i = o(n_i) \\
&= \limsup_{i\to\infty} \frac{\sum_{j=0}^{i} \left|\mathrm{enc}\left(\mathrm{ind}^{(k_j)}_{d,c_j,S\upharpoonright n_{j-1}}(s_j)\right)\pi(c_j)\right|}{n_i} \\
&= \limsup_{i\to\infty} \frac{\sum_{j=0}^{i} \log \mathrm{ind}^{(k_j)}_{d,c_j,S\upharpoonright n_{j-1}}(s_j)}{n_i} && \text{since } |\pi(c_i)| = o(n_i) \\
&\leq r. && \text{by Lemma 3.5}
\end{aligned}$$

Similarly, $\rho^-_M(S,P) \le t$. □

The next theorem is a Δ-bounded analog of Corollary 3.2. It states that Δ-strong dimension may be extracted, using Δ-Turing reductions, from any sequence of positive Δ-dimension. The proof strongly resembles the proof of Theorem 3.1, using Theorem 3.7 in place of Theorem 2.10, and using Theorem 2.7 to replace Kolmogorov complexity with space-bounded Kolmogorov complexity. Also, the reduction is not uniform: the OTMs used depend on the sequence used for the extraction.

Theorem 3.8. *For each $S \in \mathbf{C}$ such that $\dim_\Delta(S) > 0$, and each $\epsilon > 0$, there exists $P \in \mathbf{C}$ such that $P \equiv^\Delta_\mathrm{T} S$ and $\mathrm{Dim}_\Delta(P) \ge 1 - \epsilon$.*

Proof. Let S and ϵ be as in the statement of the theorem. Let $0 < \delta < \epsilon \cdot \dim_\Delta(S)/2$. Choose $P \in \mathbf{C}$ and $M \in \mathrm{OTM}$ for S and δ as in Theorem 3.7 such that $P \equiv^\Delta_\mathrm{T} S$ and, letting $p_i = \#(M^P, S \upharpoonright i)$ for all $i \in \mathbb{N}$,

$$
\begin{aligned}
&\liminf_{i\to\infty} \frac{p_i}{i} < \dim_\Delta(S) + \delta \\
\Longrightarrow\ & \inf_{q\in\mathrm{sb}(\Delta)} \liminf_{n\to\infty} \frac{\mathrm{KS}^q(S\upharpoonright n)}{n} > \liminf_{i\to\infty}\frac{p_i}{i} - \delta \qquad\qquad \text{by Theorem 2.7} \\
\Longrightarrow\ & (\forall q \in \mathrm{sb}(\Delta))\ \liminf_{n\to\infty} \frac{\mathrm{KS}^q(S\upharpoonright n)}{n} > \liminf_{i\to\infty}\frac{p_i}{i} - \delta \\
\Longrightarrow\ & (\forall q \in \mathrm{sb}(\Delta))(\exists^\infty i \in \mathbb{N})\ \liminf_{n\to\infty} \frac{\mathrm{KS}^q(S\upharpoonright n)}{n} > \frac{p_i}{i} - \delta \\
\Longrightarrow\ & (\forall q \in \mathrm{sb}(\Delta))(\exists^\infty n \in \mathbb{N})\ \mathrm{KS}^q(S\upharpoonright n) > p_n - \delta n. \qquad (3.14)
\end{aligned}
$$

Since $\delta < \epsilon \cdot \dim_\Delta(S)/2$, by Theorem 2.7,

$$(\forall q \in \mathrm{sb}(\Delta))(\forall^\infty n \in \mathbb{N})\ \mathrm{KS}^q(S\upharpoonright n) > \frac{2}{\epsilon}\delta n. \qquad (3.15)$$

Note that the construction of M in the proof of Theorem 3.7 is such that entire prefixes of the oracle sequence are queried at once: whenever the bit at index $i \in \mathbb{N}$ is queried, all bits $j < i$ are also queried. Thus, a program M' simulating M with the first p_n bits of P can calculate $S \upharpoonright n$, and M' can be encoded in $|\mathrm{enc}(P \upharpoonright p_n)| + O(1)$ bits. Thus there is a constant c and, since M uses Δ space, there exists $t \in \mathrm{sb}(\Delta)$ such that $\mathrm{KS}^t(S \upharpoonright n) \le p_n + 2\log p_n + c$, which together with (3.15) implies that

$$(\forall^\infty n \in \mathbb{N})\ \delta n < \frac{\epsilon}{2}(p_n + 2\log p_n + c). \qquad (3.16)$$

Combining (3.14) and (3.16),

$$(\forall q \in \mathrm{sb}(\Delta))(\exists^\infty n \in \mathbb{N})\ \mathrm{KS}^q(S\upharpoonright n) > p_n - \frac{\epsilon}{2}\left(p_n + 2\log p_n + c\right). \qquad (3.17)$$

Suppose for the sake of contradiction that $\mathrm{Dim}_\Delta(P) < 1 - \epsilon$. Then by Theorem 2.7, it would be the case that

$$(\exists s \in \mathrm{sb}(\Delta))(\forall^\infty m \in \mathbb{N})\ \mathrm{KS}^s(P\upharpoonright m) < m - \epsilon m. \qquad (3.18)$$

Since $\dim_\Delta(S) > 0$, S is not Δ-computable (see [Lut03a], Lemma 4.13), and since S is Δ-computable by M with access to P, p_n must grow unboundedly with n. A program that produces $P \restriction p_n$ in $s(p_n)$ space can be used in conjunction with M (which uses at most $t(n)$ space) to produce $S \restriction n$ in at most $t(n)+s(p_n)$ space. For the case $\Delta = \text{comp}$, $t(n)+s(p_n)$ is bounded by a computable function of n. By part 3 of Theorem 3.7, $p_n = O(n)$, so, for the case $\Delta = \text{p}_i\text{space}$, the space bound $n \mapsto t(n)+s(p_n)$ is contained in $G_i = \text{sb}(\text{p}_i\text{space})$. Then for a suitable constant $c' \approx |M_d|$,

$$\begin{aligned}(\exists q \in \text{sb}(\Delta))(\forall^\infty n \in \mathbb{N})\ \text{KS}^q(S \restriction n) &\leq \text{KS}^s(P \restriction p_n) + c' \\ &< p_n - \epsilon p_n + c' \\ &< p_n - \frac{\epsilon}{2}\left(p_n + 2\log p_n + c\right). \qquad (3.19)\end{aligned}$$

But (3.19) contradicts (3.17). Hence, $\text{Dim}_\Delta(P) \geq 1 - \epsilon$. □

4 Resource-Bounded Dimension Characterizations

The following characterization of the constructive dimensions of individual sequences was shown in [Dot06].

Theorem 4.1. [Dot06] *For all $S \in \mathbf{C}$,*

$$\dim(S) = \rho^-(S), \text{ and } \text{Dim}(S) = \rho^+(S).$$

For the sake of completeness, we note that an analog of Theorem 4.1 has been shown to hold in the world of finite-state machines. For a sequence $S \in \mathbf{C}$, the *finite-state dimension* $\dim_{\text{FS}}(S)$ [DLLM04] of S and the *finite-state strong dimension* $\text{Dim}_{\text{FS}}(S)$ [AHLM] of S are each defined in analogy to the Δ-dimensions, using martingales implemented by finite-state machines. $\rho^-_{\text{FS}}(S)$ and $\rho^+_{\text{FS}}(S)$ are respectively defined in analogy to $\rho^-_\Delta(S)$ and $\rho^+_\Delta(S)$, with the Δ-Turing reduction replaced by an *information lossless finite-state compressor* [Sha48, Huf59]. The following theorem was shown in [DLLM04] and [AHLM].

Theorem 4.2. [DLLM04, AHLM] *For all $S \in \mathbf{C}$,*

$$\dim_{\text{FS}}(S) = \rho^-_{\text{FS}}(S), \text{ and } \text{Dim}_{\text{FS}}(S) = \rho^+_{\text{FS}}(S).$$

Theorem 3.7 and Lemma 3.4 immediately imply the following analog of Theorems 4.1 and 4.2, which similarly characterizes the computable dimensions and $\text{p}_i$space dimensions of individual sequences.

Theorem 4.3. *For all $i \in \mathbb{N}$ and $S \in \mathbf{C}$,*

$$\begin{aligned}\dim_{\text{p}_i\text{space}}(S) &= \rho^-_{\text{p}_i\text{space}}(S), \\ \text{Dim}_{\text{p}_i\text{space}}(S) &= \rho^+_{\text{p}_i\text{space}}(S), \\ \dim_{\text{comp}}(S) &= \rho^-_{\text{tt}}(S), \\ \text{Dim}_{\text{comp}}(S) &= \rho^+_{\text{tt}}(S).\end{aligned}$$

Acknowledgments. I thank Xiaoyang Gu, Pavan Aduri, Satyadev Nandakumar, and Scott Summers for helpful discussions, and Fengming Wang for pointing out [VV02].

References

[AHLM] K. B. Athreya, J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. Effective strong dimension, algorithmic information, and computational complexity. *SIAM Journal on Computing.* To appear. Preliminary version appeared in *Proceedings of the 21st International Symposium on Theoretical Aspects of Computer Science*, pages 632-643.

[Ben88] C. H. Bennett. Logical depth and physical complexity. In R. Herken, editor, *The Universal Turing Machine: A Half-Century Survey*, pages 227–257. Oxford University Press, London, 1988.

[BFNV05] H. Buhrman, L. Fortnow, I. Newman, and N. Vereshchagin. Increasing Kolmogorov complexity. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science*, number 3404 in Lecture Notes on Computer Science, pages 412–421, Berlin, 2005. Springer.

[BLW94] R. V. Book, J. H. Lutz, and K. W. Wagner. An observation on probability versus randomness with applications to complexity classes. *Mathematical Systems Theory*, 27:201–209, 1994.

[DLLM04] J. J. Dai, J. I. Lathrop, J. H. Lutz, and E. Mayordomo. Finite-state dimension. *Theoretical Computer Science*, 310:1–33, 2004. Preliminary version appeared in *Proceedings of the 28th International Colloquium on Automata, Languages, and Programming*, pages 1028–1039, 2001.

[Dot06] D. Doty. Every sequence is decompressible from a random one. In *Logical Approaches to Computational Barriers, Proceedings of the Second Conference on Computability in Europe*, volume 3988 of *Computability in Europe*, Swansea, UK, July 2006. Lecture Notes in Computer Science. to appear.

[Edg04] G. A. Edgar. *Classics on Fractals.* Westview Press, Oxford, U.K., 2004.

[Fal03] K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications.* Wiley, second edition, 2003.

[FHP$^+$06] L. Fortnow, J. Hitchcock, A. Pavan, N. V. Vinodchandran, and F. Wang. Extracting Kolmogorov complexity with applications to dimension zero-one laws. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming.* Springer, 2006. To appear.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6:675–695, 1977.

[GL05] X. Gu and J. H. Lutz. Dimension characterizations of complexity classes. Technical Report 160, Electronic Colloqium on Computational Complexity, 2005.

[Hau19] F. Hausdorff. Dimension und äusseres Mass. *Mathematische Annalen*, 79:157–179, 1919. English version appears in [Edg04], pp. 75-99.

[Hit03] J. M. Hitchcock. *Effective fractal dimension: foundations and applications.* PhD thesis, Iowa State University, 2003.

[Huf59] D. A. Huffman. Canonical forms for information-lossless finite-state logical machines. *IRE Trans. Circuit Theory CT-6 (Special Supplement)*, pages 41–59, 1959. Also available in E.F. Moore (ed.), Sequential Machine: Selected Papers, Addison-Wesley, 1964, pages 866-871.

[Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.

[Lut93] J. H. Lutz. A pseudorandom oracle characterization of BPP. *SIAM Journal on Computing*, 22(5):1075–1086, 1993.

[Lut03a] J. H. Lutz. Dimension in complexity classes. *SIAM Journal on Computing*, 32:1236–1259, 2003. Preliminary version appeared in *Proceedings of the Fifteenth Annual IEEE Conference on Computational Complexity*, pages 158–169, 2000.

[Lut03b] J. H. Lutz. The dimensions of individual strings and sequences. *Information and Computation*, 187:49–79, 2003. Preliminary version appeared in *Proceedings of the 27th International Colloquium on Automata, Languages, and Programming*, pages 902–913, 2000.

[LV97] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Springer-Verlag, Berlin, 1997. Second Edition.

[Mar66] D. A. Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 12:295–310, 1966.

[May02] E. Mayordomo. A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Information Processing Letters*, 84(1):1–3, 2002.

[MM04] W. Merkle and N. Mihailović. On the construction of effective random sets. *Journal of Symbolic Logic*, pages 862–878, 2004.

[MN05] J. S. Miller and A. Nies. Randomness and computability: Open questions. Technical report, University of Auckland, 2005.

[NR06] A. Nies and J. Reimann. A lower cone in the wtt degrees of non-integral effective dimension. In *Proceedings of IMS workshop on Computational Prospects of Infinity*, 2006. to appear.

[Pap94] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Rei04] J. Reimann. *Computability and Fractal Dimension*. PhD thesis, Universität Heidelberg, 2004.

[Rya84] B. Ya. Ryabko. Coding of combinatorial sources and Hausdorff dimension. *Soviet Mathematics Doklady*, 30:219–222, 1984.

[Rya86] B. Ya. Ryabko. Noiseless coding of combinatorial sources. *Problems of Information Transmission*, 22:170–179, 1986.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sha02] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[Soa87] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Springer-Verlag, Berlin, 1987.

[Sul84] D. Sullivan. Entropy, Hausdorff measures old and new, and limit sets of geometrically finite Kleinian groups. *Acta Mathematica*, 153:259–277, 1984.

[Tri82] C. Tricot. Two definitions of fractional dimension. *Mathematical Proceedings of the Cambridge Philosophical Society*, 91:57–74, 1982.

[von63] J. von Neumann. Various techniques for use in connection with random digits. In *von Neumann's Collected Works*, volume 5, pages 768–770. Pergamon, 1963.

[VV02] N. K. Vereshchagin and M. V. Vyugin. Independent minimum length programs to translate between given strings. *Theoretical Computer Science*, 271(1-2):131–143, 2002.