Extracting Kolmogorov Complexity with Applications to Dimension Zero-One Laws

Lance Fortnow[1], John M. Hitchcock[2] ⋆, A. Pavan[3] ⋆⋆,
N. V. Vinodchandran[4] ⋆⋆⋆, and Fengming Wang[3] †

[1] Department of Computer Science, University of Chicago, fortnow@cs.uchicago.edu
[2] Department of Computer Science, University of Wyoming, jhitchco@cs.uwyo.edu
[3] Department of Computer Science, Iowa State University, {pavan, wfengm}@cs.iastate.edu
[4] Department of Computer Science and Engineering, University of Nebraska-Lincoln, vinod@cse.unl.edu

Abstract. We apply recent results on extracting randomness from independent sources to "extract" Kolmogorov complexity. For any $\alpha, \epsilon > 0$, given a string x with $K(x) > \alpha|x|$, we show how to use a constant number of advice bits to efficiently compute another string y, $|y| = \Omega(|x|)$, with $K(y) > (1-\epsilon)|y|$. This result holds for both classical and space-bounded Kolmogorov complexity.

We use the extraction procedure for space-bounded complexity to establish zero-one laws for polynomial-space strong dimension. Our results include:

(i) If $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}) > 0$, then $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}/O(1)) = 1$.
(ii) $\mathrm{Dim}(\mathrm{E}/O(1) \mid \mathrm{ESPACE})$ is either 0 or 1.
(iii) $\mathrm{Dim}(\mathrm{E/poly} \mid \mathrm{ESPACE})$ is either 0 or 1.

In other words, from a dimension standpoint and with respect to a small amount of advice, the exponential-time class E is either minimally complex or maximally complex within ESPACE.

1 Introduction

Kolmogorov complexity quantifies the amount of randomness in an individual string. If a string x has Kolmogorov complexity m, then x is often said to contain m bits of randomness. Given x, is it possible to compute a string of length m that is Kolmogorov-random? In general this is impossible but we do make progress in this direction if we allow a tiny amount of extra information. We give a *polynomial-time computable procedure* which takes x with an additional constant amount of advice and outputs a nearly Kolmogorov-random string whose length is linear in m. Formally, for any $\alpha, \epsilon > 0$, given a string x with $K(x) > \alpha|x|$, we

⋆ This research was supported in part by NSF grant 0515313.
⋆⋆ This research was supported in part by NSF grant 0430807.
⋆⋆⋆ This research was supported in part by NSF grant 0430991.
† This research was supported in part by NSF grant 0430807.

show how to use a constant number of advice bits to compute another string y, $|y| = \Omega(|x|)$, in polynomial-time that satisfies $K(y) > (1-\epsilon)|y|$. The number of advice bits depends only on α and ϵ, but the content of the advice depends on x. This computation needs only polynomial time, and yet it extracts unbounded Kolmogorov complexity.

Our proofs use a recent construction of extractors using multiple independent sources. Traditional extractor results [13, 22, 19, 12, 21, 15, 16, 20, 9, 18, 17, 4] show how to take a distribution with high min-entropy and some truly random bits to create a close to uniform distribution. Recently, Barak, Impagliazzo, and Wigderson [2] showed how to eliminate the need for a truly random source when several independent random sources are available. We make use of these extractors for our main result on extracting Kolmogorov complexity. Barak et al. [3] and Raz [14] have further extensions on extracting from independent sources.

To make the connection, consider the uniform distribution on the set of strings x whose Kolmogorov complexity is at most m. This distribution has min-entropy about m and x acts like a random member of this set. We can define a set of strings $x_1, \ldots, x_k$ to be independent if $K(x_1 \ldots x_k) \approx K(x_1) + \cdots + K(x_k)$. By symmetry of information this implies $K(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k) \approx K(x_i)$. Combining these ideas we are able to apply the extractor constructions for multiple independent sources to Kolmogorov complexity.

To extract the randomness from a string x, we break x into a number of substrings $x_1, \ldots, x_l$, and view each substring x_i as coming from an independent random source. Of course, these substrings may not be independently random in the Kolmogorov sense. We find it a useful concept to quantify the *dependency within* x as $\sum_{i=1}^{l} K(x_i) - K(x)$. Another technical problem is that the randomness in x may not be nicely distributed among these substrings; for this we need to use a small (constant) number of nonuniform advice bits.

This result about extracting Kolmogorov-randomness also holds for polynomial-space bounded Kolmogorov complexity. We apply this to obtain zero-one laws for the dimensions of certain complexity classes. Polynomial-space dimension [11] and strong dimension [1] have been developed to study the quantitative structure of classes that lie in E and ESPACE. These dimensions are resource-bounded versions of Hausdorff dimension and packing dimension, respectively, the two most important fractal dimensions. Polynomial-space dimension and strong dimension refine pspace-measure [10] and have been shown to be duals of each other in many ways [1]. Additionally, polynomial-space strong dimension is closely related to pspace-category [7]. In this paper we focus on polynomial-space strong dimension which quantifies PSPACE and ESPACE in the following way:

- $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{PSPACE}) = 0$.
- $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{ESPACE}) = 1$.

It is interesting to consider the dimension of a complexity class $\mathcal{C}$, where $\mathcal{C}$ is contained in ESPACE. The dimension is always a real number between zero and one inclusive. Can a reasonable complexity class have a fractional dimension? In particular consider the class E. Deciding the polynomial-space dimension of

E would imply a major complexity separation, but perhaps we can show that E must have dimension either zero or one, a "zero-one" law for dimension.

We can show such a zero-one law if we add a small amount of nonuniform advice. An equivalence between space-bounded Kolmogorov complexity rates and strong pspace-dimension allows us to use our Kolmogorov-randomness extraction procedure to show the following results.

(i) If $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}) > 0$, then $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}/O(1)) = 1$.
(ii) $\mathrm{Dim}(\mathrm{E}/O(1) \mid \mathrm{ESPACE})$ is either 0 or 1.
(iii) $\mathrm{Dim}(\mathrm{E/poly} \mid \mathrm{ESPACE})$ is either 0 or 1.

2 Preliminaries

2.1 Kolmogorov Complexity

Let M be a Turing machine. Let $f : \mathbb{N} \to \mathbb{N}$. For any $x \in \{0,1\}^*$, define

$$K_M(x) = \min\{|\pi| \mid M(\pi) \text{ prints } x\}$$

and

$$KS^f_M(x) = \min\{|\pi| \mid M(\pi) \text{ prints } x \text{ using at most } f(|x|) \text{ space}\}.$$

There is a universal machine U such that for every machine M, there is some constant c such that for all x, $K_U(x) \le K_M(x)+c$ and $KS^{cf+c}_U(x) \le KS^f_M(x)+c$ [8]. We fix such a machine U and drop the subscript, writing $K(x)$ and $KS^f(x)$, which are called the *(plain) Kolmogorov complexity of x* and *f-bounded (plain) Kolmogorov complexity of x*. While we use plain complexity in this paper, our results also hold for prefix-free complexity.

The following definition quantifies the fraction of randomness in a string.

Definition. For a string x, the *rate* of x is $rate(x) = K(x)/|x|$. For a polynomial g, the *g-rate of x* is $rate^g(x) = KS^g(x)/|x|$.

2.2 Polynomial-Space Dimension

We now review the definitions of polynomial-space dimension [11] and strong dimension [1]. For more background we refer to these papers and the recent survey paper [6].

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

For a language A, we write $A \upharpoonright n$ for the first n bits of A's characteristic sequence (according to the standard enumeration of $\{0,1\}^*$) and $A \upharpoonright [i,j]$ for the subsequence beginning from the ith bit and ending at the jth bit. An s-gale d *succeeds on* a language A if $\limsup\limits_{n\to\infty} d(A \upharpoonright n) = \infty$ and d *succeeds strongly on* A if $\liminf\limits_{n\to\infty} d(A \upharpoonright n) = \infty$. The *success set* of d is $S^\infty[d] = \{A \mid d \text{ succeeds on } S\}$. The *strong success set* of d is $S^\infty_{\mathrm{str}}[d] = \{A \mid d \text{ succeeds strongly on } S\}$.

Definition. Let X be a class of languages.

1. The pspace-*dimension* of X is

$$\dim_{\text{pspace}}(X) = \inf \left\{ s \,\middle|\, \begin{array}{l} \text{there is a polynomial-space computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^{\infty}[d] \end{array} \right\}.$$

2. The *strong* pspace-*dimension* of X is

$$\text{Dim}_{\text{pspace}}(X) = \inf \left\{ s \,\middle|\, \begin{array}{l} \text{there is a polynomial-space computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^{\infty}_{\text{str}}[d] \end{array} \right\}.$$

For every X, $0 \leq \dim_{\text{pspace}}(X) \leq \text{Dim}_{\text{pspace}}(X) \leq 1$. An important fact is that ESPACE has pspace-dimension 1, which suggests the following definitions.

Definition. Let X be a class of languages.

1. The *dimension of X within* ESPACE is

$$\dim(X \mid \text{ESPACE}) = \dim_{\text{pspace}}(X \cap \text{ESPACE}).$$

2. The *strong dimension of X within* ESPACE is

$$\text{Dim}(X \mid \text{ESPACE}) = \text{Dim}_{\text{pspace}}(X \cap \text{ESPACE}).$$

In this paper we will use an equivalent definition of the above dimensions in terms of space-bounded Kolmogorov complexity.

Definition. Given a language L and a polynomial g the *g-rate of L* is

$$\mathit{rate}^g(L) = \liminf_{n\to\infty} \mathit{rate}^g(L\restriction n).$$

strong g-rate of L is

$$\mathit{Rate}^g(L) = \limsup_{n\to\infty} \mathit{rate}^g(L\restriction n).$$

Theorem 2.1. (Hitchcock [5]) *Let* poly *denote all polynomials. For every class X of languages,*

$$\dim_{\text{pspace}}(X) = \inf_{g\in\text{poly}} \sup_{L\in X} \mathit{rate}^g(L).$$

and

$$\text{Dim}_{\text{pspace}}(X) = \inf_{g\in\text{poly}} \sup_{L\in X} \mathit{Rate}^g(L).$$

3 Extracting Kolmogorov Complexity

Barak, Impagliazzo, and Wigderson [2] recently gave an explicit multi-source extractor.

Theorem 3.1. *([2]) For every constant $0 < \sigma < 1$, and $c > 1$ there exist $l = poly(1/\sigma, c)$, a constant r and a computable function $E : \Sigma^{\ell n} \to \Sigma^n$ such that if $H_1, \cdots, H_l$ are independent distributions over Σ^n, each with min entropy at least σn, then $E(H_1, \cdots, H_l)$ is 2^{-cn}-close to U_n, where U_n is the uniform distribution over Σ^n. Moreover, E runs in time n^r.*

We show that the above extractor can be used to produce nearly Kolmogorov-random strings from strings with high enough complexity. The following notion of dependency is useful for quantifying the performance of the extractor.

Definition. Let $x = x_1 x_2 \cdots x_k$, where each x_i is an n-bit string. The *dependency within* x, $dep(x)$, is defined as $\sum_{i=1}^{k} K(x_i) - K(x)$.

Theorem 3.2. *For every $0 < \sigma < 1$ and large enough n, there exist a constant $l > 1$, and a polynomial-time computable function E such that if $x_1, x_2, \cdots x_l$ are n-bit strings with $K(x_i) \geq \sigma n$, $1 \leq i \leq l$, then*

$$K(E(x_1, \cdots, x_l)) \geq n - 10l \log n - dep(x).$$

Proof. Let $0 < \sigma' < \sigma$. By Theorem 3.1, there is a constant l and a polynomial-time computable multi-source extractor E such that if $H_1, \cdots, H_l$ are independent sources each with min-entropy at least $\sigma' n$, then $E(H_1, \cdots, H_l)$ is 2^{-5n} close to U_n.

We show that this extractor also extracts Kolmogorov complexity. We prove by contradiction. Suppose the conclusion is false, i.e,

$$K(E(x_1, \cdots x_l)) < n - 10l \log n - dep(x).$$

Let $K(x_i) = m_i$, $1 \leq i \leq l$. Define the following sets:

$$I_i = \{y \mid y \in \Sigma^n, K(y) \leq m_i\},$$

$$Z = \{z \in \Sigma^n \mid K(z) < n - 10l \log n - dep(x)\},$$

$$Small = \{\langle y_1, \cdots, y_l\rangle \mid y_i \in I_i, \text{ and } E(y_1, \cdots y_l) \in Z\}.$$

By our assumption $\langle x_1, \cdots x_l\rangle$ belongs to $Small$. We use this to arrive at a contradiction regarding the Kolmogorov complexity of $x = x_1 x_2 \cdots x_l$. We first calculate an upper bound on the size of $Small$.

Observe that the set $\{xy \mid x \in \Sigma^{\sigma' n}, y = 0^{n-\sigma' n}\}$ is a subset of each of I_i. Thus the cardinality of each of I_i is at least $2^{\sigma' n}$. Let H_i be the uniform distribution on I_i. Thus the min-entropy of H_i is at least $\sigma' n$.

Since H_i's have min-entropy at least $\sigma' n$, $E(H_1, \cdots, H_l)$ is 2^{-5n}-close to U_n. Then

$$\Big|P[E(H_1, \ldots, H_l) \in Z] - P[U_n \in Z]\Big| \leq 2^{-5n}. \tag{3.1}$$

Note that the cardinality of I_i is at most 2^{m_i+1}, as there are at most 2^{m_i+1} strings with Kolmogorov complexity at most m_i. Thus H_i places a weight of at least 2^{-m_i-1} on each string from I_i. Thus $H_1 \times \cdots \times H_l$ places a weight of at least $2^{-(m_1+\cdots+m_l+l)}$ on each element of $Small$. Therefore,

$$P[E(H_1, \ldots, H_l) \in Z] = P[(H_1, \ldots, H_l) \in Small] \geq |Small| \cdot 2^{-(m_1+\cdots+m_l+l)},$$

and since $|Z| \leq 2^{n-10l\log n - dep(x)}$, from (3.1) we obtain

$$|Small| < 2^{m_1+1} \times \cdots \times 2^{m_l+1} \times \left(\frac{2^{n-10l\log n - dep(x)}}{2^n} + 2^{-5n} \right)$$

Without loss of generality we can take $dep(x) < n$, otherwise the theorem is trivially true. Thus $2^{-5n} < 2^{-10l\log n - dep(x)}$. Using this and the fact that l is a constant independent of n, we obtain

$$|Small| < 2^{m_1+\cdots+m_l - dep(x) - 8l\log n},$$

when n is large enough. Since $K(x) = K(x_1) + \cdots + K(x_l) - dep(x)$,

$$|Small| < 2^{K(x)-8l\log n}.$$

We first observe that there is a program Q that, given the values of m_i's, n, l, and $dep(x)$ as auxiliary inputs, recognizes the set $Small$. This program works as follows: Let $z = z_1 \cdots z_l$, where $|z_i| = n$. For each program P_i of length at most m_i check whether P_i outputs z_i, by running the P_i's in a dovetail fashion. If it is discovered that for each of z_i, $K(z_i) \leq m_i$, then compute $y = E(z_1, \cdots, z_l)$. Now verify that $K(y)$ is at most $n - dep(x) - 10l\log n$. This again can be done by running programs of the length at most $n - dep(x) - 10l\log n$ in a dovetail manner. If it is discovered that $K(y)$ is at most $n - dep(x) - 10l\log n$, then accept z.

So given the values of parameters n, $dep(x)$, l and m_is, there is a program P that enumerates all elements of $Small$. Since by our assumption x belongs to $Small$, x appears in this enumeration. Let i be the position of x in this enumeration. Since $|Small|$ is at most $2^{K(x)-8l\log n}$, i can be described using $K(x) - 8l\log n$ bits.

Thus there is a program P' based on P that outputs x. This program takes i, $dep(x)$, n, $m_1, \cdots, m_l$, and l, as auxiliary inputs. Since the m_i's and $dep(x)$ are bounded by n,

$$\begin{aligned} K(x) &\leq K(x) - 8l\log n + 2\log n + l\log n + O(1) \\ &\leq K(x) - 5l\log n + O(1), \end{aligned}$$

which is a contradiction. □

If $x_1, \cdots x_l$ are independent strings with $K(x_i) \geq \sigma n$, then $E(x_1, \cdots, x_l)$ is a Kolmogorov random string of length n.

Corollary 3.3. *For every constant $0 < \sigma < 1$, there exists a constant l, and a polynomial-time computable function E such that if $x_1, \cdots x_l$ are n-bit strings such $K(x_i) \geq \sigma n$, and $K(x_1 x_2 \cdots x_l) = \sum K(x_i) - O(\log n)$, then $E(x_1, \cdots, x_l)$ is Kolmogorov random, i.e.,*

$$K(E(x_1, \cdots, x_l)) > n - O(\log n).$$

This theorem says that given $x \in \Sigma^{ln}$, if each piece x_i has high enough complexity and the dependency with x is small, then we can output a string y whose Kolmogorov rate is higher than the Kolmogorov rate of x, i.e, y is relatively more random than x. What if we only knew that x has high enough complexity but knew nothing about the complexity of individual pieces or the dependency within x? Our next theorem states that in this case also there is a procedure producing a string whose rate is higher than the rate of x. However, this procedure needs constant bits of advice.

Theorem 3.4. *For all real numbers $0 < \alpha < \beta < 1$ there exist a constant $0 < \gamma < 1$, constants $c, l, n_0 \geq 1$, and a procedure R such that the following holds. For any string x with $|x| \geq n_0$ and $rate(x) \geq \alpha$, there exists an advice string a_x such that*

$$rate(R(x, a_x)) \geq \min\{rate(x) + \gamma, \beta\}$$

where $|a_x| = c$. Moreover, R runs in polynomial time, and $|R(x, a_x)| = \lfloor |x|/l \rfloor$.

The number c depends only on α, β and is independent of x. However, the contents of a_x depend on x.

Proof. Let $\alpha' < \alpha$ and $\epsilon < \min\{1 - \beta, \alpha'\}$. Let $\sigma = (1 - \epsilon)\alpha'$. Using parameter σ in Theorem 3.2, we obtain a constant $l > 1$ and a polynomial-time computable function E that extracts Kolmogorov complexity.

Let $\beta' = 1 - \frac{\epsilon}{2}$, and $\gamma = \frac{\epsilon^2}{2l}$. Observe that $\gamma \leq \frac{1-\beta'}{l}$ and $\gamma < \frac{\alpha' - \sigma}{l}$.

Let x have $rate(x) = \nu \geq \alpha$. Let $n, k \geq 0$ such that $|x| = ln + k$ and $k < l$. We strip the last k bits from x and write $x = x_1 \cdots x_l$ where each $|x_i| = n$. Let $\nu' = rate(x)$ after this change. We have $\nu' > \nu - \gamma/2$ and $\nu' > \alpha'$ if $|x|$ is sufficiently large.

We consider three cases.

Case 1. There exists j, $1 \leq j \leq l$ such that $K(x_j) < \sigma n$.
Case 2. Case 1 does not hold and $dep(x) \geq \gamma ln$.
Case 3. Case 1 does not hold and $dep(x) < \gamma ln$.

We have two claims about Cases 1 and 2:

Claim 3.5. Assume Case 1 holds. There exists i, $1 \leq i \leq l$, such that $rate(x_i) \geq \nu' + \gamma$.

Proof of Claim 3.5. Suppose not. Then for every $i \neq j$, $1 \leq i \leq l$, $K(x_i) \leq (\nu' + \gamma)n$. We can describe x by describing x_j which takes σn bits, and all the x_i's, $i \neq j$. Thus the total complexity of x would be at most

$$(\nu' + \gamma)(l - 1)n + \sigma n + O(\log n)$$

Since $\gamma < \frac{\alpha' - \sigma}{l}$ and $\alpha' < \nu'$ this quantity is less than $\nu' ln$. Since the rate of x is ν', this is a contradiction. $\square$ *Claim 3.5*

Claim 3.6. Assume Case 2 holds. There exists i, $1 \leq i \leq l$, $rate(x_i) \geq \nu' + \gamma$.

Proof of Claim 3.6. By definition,

$$K(x) = \sum_{i=1}^{l} K(x_i) - dep(x)$$

Since $dep(x) \geq \gamma ln$ and $K(x) \geq \nu' ln$,

$$\sum_{i=1}^{l} K(x_i) \geq (\nu' + \gamma) ln.$$

Thus there exists i such that $rate(x_i) \geq \nu' + \gamma$. $\square$ *Claim 3.6*

We can now describe the constant number of advice bits. The advice a_x contains the following information: which of the three cases described above holds, and

- If Case 1 holds, then from Claim 3.5 the index i such that $rate(x_i) \geq \nu' + \gamma$.
- If Case 2 holds, then from Claim 3.6 the index i such that $rate(x_i) \geq \nu' + \gamma$.

Since $1 \leq i \leq l$, the number of advice bits is bounded by $O(\log l)$. We now describe procedure R. When R takes an input x, it first examines the advice a_x. If Case 1 or Case 2 holds, then R simply outputs x_i. Otherwise, Case 3 holds, and R outputs $E(x)$. Since E runs in polynomial time, R runs in polynomial time.

If Case 1 or Case 2 holds, then

$$rate(R(x, a_x)) \geq \nu' + \gamma \geq \nu + \tfrac{\gamma}{2}.$$

If Case 3 holds, we have $R(x, a_x) = E(x)$ and by Theorem 3.2, $K(E(x)) \geq n - 10\log n - \gamma ln$. Since $\gamma \leq \frac{1-\beta'}{l}$, in this case

$$rate(R(x, a_x)) \geq \beta' - \tfrac{10 \log n}{n}.$$

For large enough n, this value is at least β. Therefore in all three cases, the rate increases by at least $\gamma/2$ or reaches β. $\square$

We now prove our main theorem.

Theorem 3.7. *Let α and β be constants with $0 < \alpha < \beta < 1$. There exist a polynomial-time procedure $P(\cdot, \cdot)$ and constants C_1, C_2, n_1 such that for every x with $|x| \geq n_1$ and $rate(x) \geq \alpha$ there exists a string a_x with $|a_x| = C_1$ such that*

$$rate(P(x, a_x)) \geq \beta$$

and $|P(x, a_x)| \geq |x|/C_2$.

Proof. We apply the procedure R from Theorem 3.4 iteratively. Each application of R outputs a string whose rate is at least β or is at least γ more than the rate of the input string. Applying R at most $k = \lceil(\beta - \alpha)/\gamma\rceil$ times, we obtain a string whose rate is at least β.

Note that $R(y, a_y)$ has output length $|R(y, a_y)| = \lfloor |y|/l \rfloor$ and increases the rate of y if $|y| \geq n_0$. If we take $n_1 = (n_0+1)kl$, we ensure that in each application of R we have a string whose length is at least n_0. Each iteration of R requires c bits of advice, so the total number of advice bits needed is $C_1 = kc$. Thus C_1 depends only on α and β. Each application of R decreases the length by a constant fraction, so there is a constant C_2 such that the length of the final outputs string is at least $|x|/C_2$. $\square$

The proofs in this section also work for space-bounded Kolmogorov complexity. For this we need a space-bounded version of dependency.

Definition. Let $x = x_1x_2 \cdots x_k$ where each x_i is an n-bit string, let f and g be two space bounds. The (f, g)-*bounded dependency within* x, $dep_g^f(x)$, is defined as $\sum_{i=1}^{k} KS^g(x_i) - KS^f(x)$.

We obtain the following version of Theorem 3.2.

Theorem 3.8. *For every polynomial g there exists a polynomial f such that for every $0 < \sigma < 1$, there exist a constant $l > 1$, and a polynomial-time computable function E such that if $x_1, \cdots, x_l$ are n-bit strings with $KS^f(x_i) \geq \sigma n$, $1 \leq i \leq l$, then*

$$KS^g(E(x_1, \cdots, x_l)) \geq n - 10l \log n - dep_g^f(x).$$

Similarly we obtain the following extension of Theorem 3.7.

Theorem 3.9. *Let g be a polynomial and let α and β be constants with $0 < \alpha < \beta < 1$. There exist a polynomial f, polynomial-time procedure $R(\cdot, \cdot)$, and constants C_1, C_2, n_1 such that for every x with $|x| \geq n_1$ and $rate^f(x) \geq \alpha$ there exists a string a_x with $|a_x| = C_1$ such that*

$$rate^g(R(x, a_x)) \geq \beta$$

and $|R(x, a_x)| \geq |x|/C_2$.

4 Zero-One Laws

In this section we establish zero-one laws for the dimensions of certain classes within ESPACE. Our most basic result is the following, which says that if E has positive dimension, then the class $E/O(1)$ has maximal dimension.

Theorem 4.1. *If* $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}) > 0$*, then* $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}/O(1)) = 1$.

For the theorem we use the following lemma, which can be proved using Theorem 3.9. We omit the proof due to space constraints.

Lemma 4.2. *Let g be any polynomial and α, θ be rational numbers with $0 < \alpha < \theta < 1$. Then there is a polynomial f such that if there exists $L \in \mathrm{E}$ with $\mathit{Rate}^f(L) \geq \alpha$, then there exists $L' \in \mathrm{E}/O(1)$ with $\mathit{Rate}^g(L') \geq \theta$.*

Proof of Theorem 4.1. We will show that for every polynomial g, and real number $0 < \theta < 1$, there is a language L' in $\mathrm{E}/O(1)$ with $\mathit{Rate}^g(L) \geq \theta$. By Theorem 2.1, this will show that the strong pspace-dimension of $\mathrm{E}/O(1)$ is 1.

The assumption states that the strong pspace-dimension of E is greater than 0. If the strong pspace-dimension of E is actually one, then we are done. If not, let α be a positive rational number that is less than $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E})$. By Theorem 2.1, for every polynomial f, there exists a language $L \in \mathrm{E}$ with $\mathit{Rate}^f(L) \geq \alpha$.

By Lemma 4.2, from such a language L we obtain a language L' in $\mathrm{E}/O(1)$ with $\mathit{Rate}^g(L') \geq \theta$. Thus the strong pspace-dimension of $\mathrm{E}/O(1)$ is 1. □

Observe that in the above construction, if the original language L is in $\mathrm{E}/O(1)$, then also L' is in $\mathrm{E}/O(1)$, and similarly membership in E/poly is preserved. Additionally, if $L \in \mathrm{ESPACE}$, it can be shown that $L' \in \mathrm{ESPACE}$. With these observations, we obtain the following zero-one laws.

Theorem 4.3. *Each of the following is either 0 or 1.*

1. $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}/O(1))$.
2. $\mathrm{Dim}_{\mathrm{pspace}}(\mathrm{E}/\mathrm{poly})$.
3. $\mathrm{Dim}(\mathrm{E}/O(1) \mid \mathrm{ESPACE})$.
4. $\mathrm{Dim}(\mathrm{E}/\mathrm{poly} \mid \mathrm{ESPACE})$.

We remark that in Theorems 4.1 and 4.3, if we replace E by EXP, the theorems still hold. The proofs also go through for other classes such as BPEXP, $\mathrm{NEXP} \cap \mathrm{coNEXP}$, and NEXP/poly.

Theorems 4.1 and 4.3 concern strong dimension. For dimension, the situation is more complicated. Using similar techniques, we can prove that if $\mathrm{dim}_{\mathrm{pspace}}(\mathrm{E}) > 0$, then $\mathrm{dim}_{\mathrm{pspace}}(\mathrm{E}/O(1)) \geq 1/2$. Analogously, we can obtain zero-half laws for the pspace-dimension of E/poly, etc.

Acknowledgments

We thank Xiaoyang Gu and Philippe Moser for several helpful discussions.

References

1. K. B. Athreya, J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. Effective strong dimension in algorithmic information and computational complexity. *SIAM Journal on Computing*. To appear.
2. B. Barak, R. Impagliazzo, and A. Wigderson. Extracting randomness using few independent sources. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393. IEEE Computer Society, 2004.

3. B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating independence: new constructions of condensers, ramsey graphs, dispersers, and extractors. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 1–10, 2005.
4. B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. In *Proceedings of the 26th Annual IEEE Conference on Foundations of Computer Science*, pages 429–442, 1985.
5. J. M. Hitchcock. *Effective Fractal Dimension: Foundations and Applications*. PhD thesis, Iowa State University, 2003.
6. J. M. Hitchcock, J. H. Lutz, and E. Mayordomo. The fractal geometry of complexity classes. *SIGACT News*, 36(3):24–38, September 2005.
7. J. M. Hitchcock and A. Pavan. Resource-bounded strong dimension versus resource-bounded category. *Information Processing Letters*, 95(3):377–381, 2005.
8. M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag, Berlin, 1997. Second Edition.
9. C-J. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to a constant factor. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 602–611, 2003.
10. J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.
11. J. H. Lutz. Dimension in complexity classes. *SIAM Journal on Computing*, 32(5):1236–1259, 2003.
12. N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 42(2):149–167, 1999.
13. N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.
14. R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 11–20, 2005.
15. O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In *Proceedings of the 41st Annual Conference on Foundations of Computer science*, 2000.
16. O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings of the 41st Annual IEEE Conference on Foundations of Computer Science*, 2000.
17. M. Santha and U. Vazirani. Generating quasi-random sequences from slightly random sources. In *Proceedings of the 25th Annual IEEE Conference on Foundations of Computer Science*, pages 434–440, 1984.
18. R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *Proceedings of the 42nd Annual Conference on Foundations of Computer Science*, 2001.
19. A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999.
20. A. Ta-Shma, D. Zuckerman, and M. Safra. Extractors from reed-muller codes. In *Proceedings of the 42nd Annual Conference on Foundations of Computer Science*, 2001.
21. L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(1):860–879, 2001.
22. D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.