Rate Distortion and Denoising of Individual Data Using Kolmogorov Complexity

Nikolai K. Vereshchagin and Paul M.B. Vitányi

***Abstract*—We examine the structure of families of distortion balls from the perspective of Kolmogorov complexity. Special attention is paid to the canonical rate-distortion function of a source word which returns the minimal Kolmogorov complexity of all distortion balls containing that word subject to a bound on their cardinality. This canonical rate-distortion function is related to the more standard algorithmic rate-distortion function for the given distortion measure. Examples are given of list distortion, Hamming distortion, and Euclidean distortion. The algorithmic rate-distortion function can behave differently from Shannon's rate-distortion function. To this end, we show that the canonical rate-distortion function can and does assume a wide class of shapes (unlike Shannon's); we relate low algorithmic mutual information to low Kolmogorov complexity (and consequently suggest that certain aspects of the mutual information formulation of Shannon's rate-distortion function behave differently than would an analogous formulation using algorithmic mutual information); we explore the notion that low Kolmogorov complexity distortion balls containing a given word capture the interesting properties of that word (which is hard to formalize in Shannon's theory) and this suggests an approach to denoising.**

***Index Terms*—Algorithmic rate distortion, characterization, denoising, distortion families, fitness of destination words, individual data, Kolmogorov complexity, rate distortion, shapes of curves.**

I. Introduction

Rate distortion theory analyzes the transmission and storage of information at insufficient bit rates. The aim is to minimize the resulting information loss expressed in a given distortion measure. The original data is called the "source word" and the encoding used for transmission or storage is called the "destination word." The number of bits available for a destination word is called the "rate." The choice of distortion measure is usually a selection of which aspects of the source word are relevant in the setting at hand, and which aspects are irrelevant (such as noise). For example, in application to lossy compression of a sound file this results in a compressed file where, among others, the very high and very low inaudible frequencies have been suppressed. The distortion measure is chosen such that it penalizes the deletion of the inaudible frequencies but lightly because they are not relevant for the auditory experience. We study rate distortion of individual source words using Kolmogorov complexity and show how it is related to denoising. The classical probabilistic theory is reviewed in Appendix A. Computability notions are reviewed in Appendix B and Kolmogorov complexity in Appendix C. Randomness deficiency according to Definition 8 and its relation to the fitness of a destination word for a source word is explained further in Appendix D. Appendix E gives the proof, required for a Hamming distortion example, that every large Hamming ball can be covered by a small number of smaller Hamming balls (each of equal cardinality). More specifically, the number of covering balls is close to the ratio between the cardinality of the large Hamming ball and the small Hamming ball. The proofs of the theorems are deferred to Appendix F.

A. Related Work

In [8], Kolmogorov formulated the "structure function" which can be viewed as a proposal for nonprobabilistic model selection. This function and the associated Kolmogorov sufficient statistic are partially treated in [19], [24], and [6] and analyzed in detail in [22]. We will show that the structure function approach can be generalized to give an approach to rate distortion and denoising of individual data.

Classical rate-distortion theory was initiated by Shannon in [17]. In [18] Shannon gave a nonconstructive asymptotic characterization of the expected rate-distortion curve of a random variable (Theorem 4 in Appendix A). References [1], [2] treat more general distortion measures and random variables in the Shannon framework.

References [25], [13], and [20] relate the classical and algorithmic approaches according to traditional information-theoretic concerns. We follow their definitions of the rate-distortion function. The results in the references show that if the source word is obtained from random i.i.d. sources, then with high probability and in expectation its individual rate-distortion curve is close to the Shannon's single rate-distortion curve. In contrast, our Theorem 1 shows that for distortion measures satisfying properties 1 through 4 there are many different shapes of individual rate-distortion functions related to the different individual source words, and many of them are very different from Shannon's rate-distortion curve.

Manuscript received December 16, 2008; revised February 24, 2010. Current version published June 16, 2010. This work was supported in part by the Russian Federation Basic Research Fund by Grants 03-01-00475, 02-01-22001, and 358.2003.1. The work of N. K. Vereshchagin was done in part while visiting CWI and was supported in part by the Russian Federation Basic Research Fund and by a visitors grant of NWO, Grant 09-01-00709. The work of P. M. B. Vitányi was supported in part by the BSIK Project BRICKS of the Dutch government and NWO, and by the EU NoE Pattern Analysis, Statistical Modeling, and Computational Learning (PASCAL). The material in this paper was presented in part at the IEEE International Symposium on Information Theory (ISIT), Seattle, WA, July 2006.

N. K. Vereshchagin is with the Department of Mathematical Logic and Theory of Algorithms, Faculty of Mechanics and Mathematics, Moscow State University, Moscow, Russia 119992 (e-mail: nikolay.vereshchagin@gmail.com).

P. M. B. Vitányi is with the CWI, Science Park 123, 1098XG Amsterdam, The Netherlands (e-mail: Paul.Vitanyi@cwi.nl).

Communicated by I. Kontoyiannis, Associate Editor for Shannon Theory.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2010.2048491

Also Ziv [26] considers a rate-distortion function for individual data. The rate-distortion function is assigned to every infinite sequence ω of letters of a finite alphabet Γ. The source words x are prefixes of ω and the encoding function is computed by a finite state transducer. Kolmogorov complexity is not involved.

In [16], [12], [4], and [5] alternative approaches to denoising via compression and in [15] and [14] applications of the current work are given.

In [22], Theorems 1 and 3 were obtained for a particular distortion measure relevant to model selection (the example $\mathcal{L}$ in this paper). The techniques used in that paper do not generalize to prove the current theorems which concern arbitrary distortion measures satisfying certain properties given here.

B. Results

A source word is taken to be a finite binary string. Destination words are finite objects (not necessarily finite binary strings). For every destination word encoding a particular source word with a certain distortion, there is a finite set of source words that are encoded by this destination word with at most that distortion. We call these finite sets of source words "distortion balls." Our approach is based on the Kolmogorov complexity of distortion balls. For every source word we define its "canonical" rate-distortion function, from which the algorithmic rate-distortion function of that source word can be obtained by a simple transformation, Lemma 2.

We assume that a distortion measure satisfies certain properties which are specified in the theorems concerned. In Theorem 1, it is shown that there are distinct canonical rate-distortion curves (and hence distinct rate-distortion curves) associated with distinct source words (although some curves may coincide). Moreover, every candidate curve from a given family of curves is realized approximately as the canonical rate-distortion curve (and, hence, for a related family of curves every curve is realized approximately as the rate-distortion curve) of some source word. In Theorem 2, we prove a Kolmogorov complexity analog for Shannon's theorem, Theorem 4 in Appendix A, on the characterization of the expected rate-distortion curve of a random variable. The new theorem states approximately the following: For every source word and every destination word there exists another destination word that has Kolmogorov complexity equal to the algorithmic information in the first destination word about the source word, up to a logarithmic additive term, and both destination words incur the same distortion with the source word. (The theorem is given in the distortion-ball formulation of destination words.) In Theorem 3 we show that, at every rate, the destination word incurring the least distortion is in fact the "best-fitting" among all destination words at that rate. "Best-fitting" is taken in the sense of sharing the most properties with the source word. (This notion of a best-fitting destination word for a source word can be expressed in Kolmogorov complexity, but not in the classic probabilistic framework. Hence there is no classical analogue for this theorem.) It turns out that this yields a method of denoising by compression.

II. Preliminaries

A. Data and Binary Strings

We write *string* to mean a finite binary string. Other finite objects can be encoded into strings in natural ways. The set of strings is denoted by $\{0,1\}^*$. The *length* of a string x is the number of bits in it denoted as $|x|$. The *empty* string ϵ has length $|\epsilon| = 0$. Identify the natural numbers $\mathcal{N}$ (including 0) and $\{0,1\}^*$ according to the correspondence

$$(0,\epsilon),(1,0),(2,1),(3,00),(4,01),\ldots. \tag{1}$$

Then, $|010| = 3$. The emphasis is on binary sequences only for convenience; observations in every finite alphabet can be so encoded in a way that is "theory neutral." For example, if a finite alphabet Σ has cardinality 2^k, then every element $i \in \Sigma$ can be encoded by $\sigma(i)$ which is a block of bits of length k. With this encoding every $x \in \Sigma^*$ satisfies that the Kolmogorov complexity $C(x) = C(\sigma(x))$ (see Appendix C for basic definitions and results on Kolmogorov complexity) up to an additive constant that is independent of x.

B. Rate-Distortion Vocabulary

Let $\mathcal{X}$ be a set, called the *source alphabet* whose elements are called *source words* or *messages*. We also use a set $\mathcal{Y}$ called the *destination alphabet*, whose elements are called *destination words*. (The destination alphabet is also called the reproduction alphabet.) In general there are no restrictions on the set $\mathcal{X}$; it can be countable or uncountable. However, for technical reasons, we assume $\mathcal{X} = \{0,1\}^*$. On the other hand, it is important that the set $\mathcal{Y}$ consists of *finite objects*: we need that the notion of Kolmogorov complexity $C(y)$ be defined for all $y \in \mathcal{Y}$. (Again, for basic definitions and results on Kolmogorov complexity see Appendix C.) In this paper it is not essential whether we use plain Kolmogorov complexity or the prefix variant; to be definite, we use plain Kolmogorov complexity.

Suppose we want to communicate a source word $x \in \mathcal{X}$ using a *destination word* $y \in \mathcal{Y}$ that can be encoded in at most r bits in the sense that the Kolmogorov complexity $C(y) \leq r$. Assume furthermore that we are given a *distortion* function $d : \mathcal{X} \times \mathcal{Y} \to \mathcal{R} \bigcup \{\infty\}$, that measures the fidelity of the destination word against the source word. Here $\mathcal{R}$ denotes the nonnegative real numbers,

Definition 1: Let $x \in \mathcal{X} = \{0,1\}^*$ and $\mathcal{Q}$ denote the rational numbers. The *rate-distortion function* $r_x : \mathcal{Q} \to \mathcal{N}$ is the minimum number of bits in a destination word y to obtain a distortion of at most δ defined by

$$r_x(\delta) = \min_{y \in \mathcal{Y}} \{C(y) : d(x,y) \leq \delta\}.$$

The "inverse" of the above function is the *distortion-rate function* $d_x : \mathcal{N} \to \mathcal{R}$ and is defined by

$$d_x(r) = \min_{y \in \mathcal{Y}} \{d(x,y) : C(y) \leq r\}.$$

These functions are analogs for individual source words x of the Shannon's rate-distortion function defined in (8) and its related distortion-rate function, expressing the least expected rate or distortion at which outcomes from a random source X can be transmitted, see Appendix A.

C. Canonical Rate-Distortion Function

Let $\mathcal{X} = \{0,1\}^*$ be the source alphabet, $\mathcal{Y}$ a destination alphabet, and d a distortion measure.

Definition 2: A *distortion ball* $B(y, \delta)$ centered on $y \in \mathcal{Y}$ with radius $\delta \in \mathcal{Q}$ is defined by

$$B(y,\delta) = \{x \in \mathcal{X} : d(x,y) \le \delta\},$$

and its cardinality is denoted by $b(y,\delta) = |B(y,\delta)|$. (We will consider only pairs $(\mathcal{Y}, d)$ such that all distortion balls are finite.) If the cardinality $b(y,\delta)$ depends only on δ but not on the center y, then we denote it by $b(\delta)$. The family $\mathcal{A}^{d,\mathcal{Y}}$ is defined as the set of all nonempty distortion balls. The restriction to strings of length n is denoted by $\mathcal{A}_n^{d,\mathcal{Y}}$.

To define the canonical rate-distortion function we need the notion of the Kolmogorov complexity of a finite set.

Definition 3: Fix a computable total order on the set of all strings [say the order defined in (1)]. The *Kolmogorov complexity $C(A)$ of a finite set* is defined as the length of the shortest string p such that the universal reference Turing machine U given p as input prints the list of all elements of A in the fixed order and halts. We require that the constituent elements are distinguishable so that we can tell them apart. Similarly we define the *conditional* versions $C(A \,|\, z)$ and $C(z \,|\, A)$ where A is a finite set of strings and z is a string or a finite set of strings.

Remark 1: In Definition 3, it is important that $U(p)$ halts after printing the last element in the list—in this way we know that the list is complete. If we allowed $U(p)$ to not halt, then we would obtain the complexity of the so-called *implicit description* of A, which can be much smaller than $C(A)$. ◊

Remark 2: We can allow $U(p)$ to output the list of elements in any order in Definition 3. This flexibility decreases $C(A)$ by at most a constant not depending on A but only depending on the order in (1). The same applies to $C(A \,|\, z)$. On the other hand, if A occurs in a conditional, such as in $C(z \,|\, A)$, then it *is* important that elements of A are given in the fixed order. This is the case since the order in which the elements of A are listed can provide extra information. ◊

Definition 4: Fix a computable bijection ϕ from the family of all finite subsets of $\{0,1\}^*$ to $\{0,1\}^*$. Let $\mathcal{A}$ be a finite family of finite subsets of $\mathcal{X} = \{0,1\}^*$. Define the *Kolmogorov complexity* $C(\mathcal{A})$ by $C(\mathcal{A}) = C(\{\phi(A)) : A \in \mathcal{A}\})$.

Remark 3: An equivalent definition of $C(A \,|\, z)$ and $C(z \,|\, A)$ as in Definition 3 is as follows. Let ϕ be as in Definition 4. Then we can define $C(A \,|\, z)$ by $C(\phi(A) \,|\, z)$ and $C(z \,|\, A)$ by $C(z \,|\, \phi(A))$. ◊

Definition 5: For every string x the *canonical rate-distortion function* $g_x : \mathcal{N} \to \mathcal{N}$ is defined by

$$g_x(l) = \min_{B \in \mathcal{A}^{d,\mathcal{Y}}} \{C(B) : x \in B, \log |B| \le l\}.$$

In a similar way, we can define the *canonical distortion-rate function*

$$h_x(j) = \min_{B \in \mathcal{A}^{d,\mathcal{Y}}} \{\log |B| : x \in B,\ C(B) \le j\}.$$

Definition 6: A *distortion family* $\mathcal{A}$ is a set of finite nonempty subsets of the set of source words $\mathcal{X} = \{0,1\}^*$. The restriction to source words of length n is denoted by $\mathcal{A}_n$.

Every destination alphabet $\mathcal{Y}$ and distortion measure d gives rise to a set of distortion balls $\mathcal{A}^{d,\mathcal{Y}}$, which is a distortion family. Thus the class of distortion families obviously includes every family of distortion balls (or distortion spheres, which is sometimes more convenient) arising from every combination of destination set and distortion measure. It is easy to see that we also can substitute the more general distortion families $\mathcal{A}$ for $\mathcal{A}^{d,\mathcal{Y}}$ in the definitions of the canonical rate-distortion and distortion-rate function. In general, the canonical rate-distortion function of x can be quite different from the rate-distortion function of x. However, by Lemma 2 it turns out that for every distortion measure satisfying certain conditions and for every x the rate-distortion function r_x is obtained from g_x by a simple transformation requiring the cardinality of the distortion balls.

Remark 4: Fix a string $x \in \mathcal{X} = \{0,1\}^*$ and consider different distortion families $\mathcal{A}$. Let $g_x^{\mathcal{A}}$ denote the canonical rate-distortion function of x with respect to a family $\mathcal{A}$. Obviously, if $\mathcal{A} \subset \mathcal{B}$ then $g_x^{\mathcal{A}}$ is pointwise not less than $g_x^{\mathcal{B}}$ (and it may happen that $g_x^{\mathcal{A}}(i) \gg g_x^{\mathcal{B}}(i)$ for some i). But as long as $\mathcal{A}$ satisfies certain natural properties, then the set of all possible g_x, when x ranges over $\mathcal{X}$, does not depend on the particular $\mathcal{A}$ involved, see Theorem 1. ◊

D. Use of the Big O Term

In the sequel we use "additive constant c" or equivalently "additive $O(1)$ term" to mean a constant. accounting for the length of a fixed binary program, independent from every variable or parameter in the expression in which it occurs. Similarly we use "$O(f(m,n,\ldots))$" to mean a function $g(m,n,\ldots)$ such that $g(m,n,\ldots) \le cf(m,n,\ldots) + c$ where c is a fixed constant independent from every variable $m, n, \ldots$ in the expression.

III. Distortion Measures

Since every family of distortion balls is a distortion family, considering arbitrary distortion measures and destination alphabets results in distortion families. We consider the following mild conditions on distortion families $\mathcal{A}$:

Property 1. For every natural number n, the family $\mathcal{A}$ contains the set $\{0,1\}^n$ of all strings of length n as an element.

Property 2. All $x, y \in A \in \mathcal{A}$ satisfy $|x| = |y|$.

Property 3. Recall that $\mathcal{A}_n = \{A \in \mathcal{A} : A \subseteq \{0,1\}^n\}$. Then, $C(\mathcal{A}_n) = O(\log n)$.

> **Property 4**. For every natural n, let α_n denote the minimal number that satisfies the following. For every positive integer c every set $A \in \mathcal{A}_n$ can be covered by at most $\alpha_n|A|/c$ sets $B \in \mathcal{A}$ with $|B| \leq c$. Call α_n the *covering coefficient* related to $\mathcal{A}_n$. Property 4 is satisfied if α_n be bounded by a polynomial in n. The smaller the covering coefficient is, the more accurate will be the description that we obtain of the shapes of the structure functions below.

The following three example families $\mathcal{A}$ satisfy all four properties.

Example 1: $\mathcal{L}$ the list distortion family. Let $\mathcal{L}_n$ be the family of all nonempty subsets of $\{0,1\}^n$. This is the family of distortion balls for list distortion, which we define as follows. Let $\mathcal{X} = \{0,1\}^*$ and $\mathcal{Y} = \bigcup_n \mathcal{L}_n$. A source word $x \in \{0,1\}^n$ is encoded by a destination word which is a subset or *list* $S \subseteq \{0,1\}^n$ with $x \in S$. Given S, we can retrieve x by its index of $\log|S|$ bits in S, ignoring rounding up, hence the name "list code." The distortion measure is $d(x,S) = \log|S|$ if $x \in S$, and ∞ otherwise. Thus, distortion balls come only in the form $B(S, \log|S|)$ with cardinality $b(S, \log|S|) = |S|$. Trivially, the covering coefficient as defined in property 4, for the list distortion family $\mathcal{L}$, satisfies $\alpha_n \leq 2$. Reference [22] describes all possible canonical distortion-rate curves, called Kolmogorov's structure function there and first defined in [8]. The distortion-rate function for list distortion coincides with the canonical distortion-rate function. The rate-distortion function of x for list distortion is

$$r_x(\delta) = \min_{S \subseteq \{0,1\}^n} \{C(S) : x \in S, \log|S| \leq \delta\}$$

and essentially coincides with the canonical rate-distortion function (g_x is the restriction of r_x to $\mathcal{N}$). ◊

Example 2: $\mathcal{H}$ the Hamming distortion family. Let $\mathcal{X} = \mathcal{Y} = \{0,1\}^*$. A source word $x \in \{0,1\}^n$ is encoded by a destination word $y \in \{0,1\}^n$. For every positive integer n, the *Hamming distance* between two strings $x = x_1 \dots x_n$ and $y = y_1 \dots y_n$ is defined by

$$d(x,y) = \frac{1}{n}|\{i : x_i \neq y_i\}|. \tag{2}$$

If x and y have different lengths, then $d(x,y) = \infty$. A *Hamming ball* in $\{0,1\}^n$ with center $y \in \{0,1\}^n$ and radius $\delta (0 \leq \delta \leq 1)$ is the set $B(y,\delta) = \{x \in \{0,1\}^n : d(x,y) \leq \delta\}$. Every x is in either $B(00\dots0, \frac{1}{2})$ or $B(11\dots1, \frac{1}{2})$, so we need to consider only Hamming distance $0 \leq \delta \leq \frac{1}{2}$. Let $\mathcal{H}_n$ be the family of all Hamming balls in $\{0,1\}^n$. We will use the following approximation of $b(\delta)$—the cardinality of Hamming balls in $\mathcal{H}_n$ of radius δ. Suppose that $0 \leq \delta \leq \frac{1}{2}$ and δn is an integer, and let $H(\delta) = \delta \log 1/\delta + (1-\delta)\log 1/(1-\delta)$ be Shannon's binary entropy function. Then,

$$2^{nH(\delta) - \log n/2 - O(1)} \leq b(\delta) \leq 2^{nH(\delta)}. \tag{3}$$

In Appendix E it is shown that the covering coefficient as defined in property 4, for the Hamming distortion family $\mathcal{H}_n$, satisfies $\alpha_n = n^{O(1)}$. The function

$$r_x(\delta) = \min_{y \in \{0,1\}^n} \{C(y) : d(x,y) \leq \delta\}$$

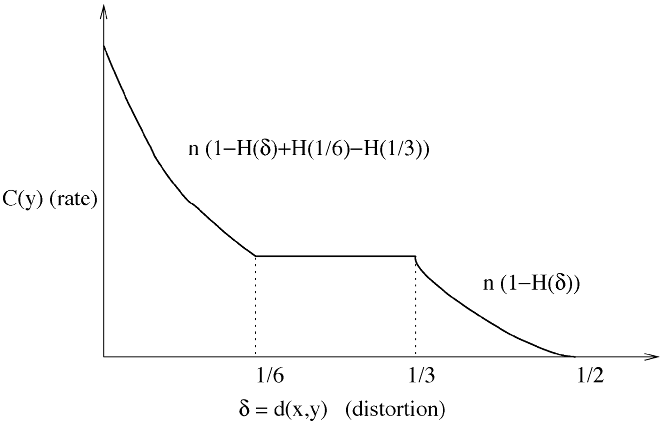

Fig. 1. Approximate rate-distortion function for Hamming distortion.

is the rate-distortion function of x for Hamming distortion. An approximation to one such function is depicted in Fig. 1. ◊

Example 3: $\mathcal{E}$ the Euclidean distortion family. Let $\mathcal{E}_n$ be the family of all intervals in $\{0,1\}^n$, where an interval is a subset of $\{0,1\}^n$ of the form $\{x : a \leq x \leq b\}$ and $\leq$ denotes the lexicographic ordering on $\{0,1\}^n$. Let $\mathcal{Y} = \{0,1\}^*$. A source word $x \in \{0,1\}^n$ is encoded by a destination word $y \in \{0,1\}^n$. Interpret strings in $\{0,1\}^n$ as binary notations for rational numbers in the segment $[0,1]$. Consider the Euclidean distance $|x-y|$ between rational numbers x and y. The balls in this metric are intervals; the cardinality of a ball of radius δ is about $\delta 2^n$. Trivially, the covering coefficient as defined in property 4, for the Euclidean distortion family $\mathcal{E}_n$, satisfies $\alpha_n \leq 2$. The function

$$r_x(\delta) = \min_{y \in \{0,1\}^n} \{C(y) : |x-y| \leq \delta\}$$

is the rate-distortion function of x for Euclidean distortion. ◊

All the properties 1 through 4 are straightforward for all three families, except property 4 in the case of the family of Hamming balls.

IV. Shapes

The rate-distortion functions of the individual strings of length n can assume roughly every shape. That is, every shape derivable from a function in the large family G_n of Definition 5 below through transformation (4).

We start the formal part of this section. Let $\mathcal{A}$ be a distortion family satisfying properties 1 through 4.

Property 1 implies that $\{0,1\}^n \in \mathcal{A}$ and property 4 applied to $\{0,1\}^n$ and $c = 1$, for every n, implies trivially that the family $\mathcal{A}$ contains the singleton set $\{x\}$ for every $x \in \{0,1\}^*$. Hence

$$g_x(0) = C(\{x\}) = C(x) + O(1).$$

Property 1 implies that for every n and string x of length n,

$$g_x(n) \leq C(\{0,1\}^n) = C(n) + O(1) \leq \log n + O(1).$$

Together this means that for every n and every string x of length n, the function $g_x(l)$ decreases from about $C(x)$ to about 0 as l increases from 0 to n.

Lemma 1: Let $\mathcal{A}$ be a distortion family satisfying properties 1 through 4. For every n and every string x of length n we have $g_x(n) = O(\log n)$, and $0 \leq g_x(l) - g_x(m) \leq m - l + O(\log n)$ for all $l < m \leq n$.

Proof: The first equation and the left-hand inequality of the second equation are straightforward. To prove the right-hand inequality, let A witness $g_x(m) = k$, which implies that $C(A) = k$ and $\log |A| \leq m$. By Property 4 there is a covering of A by at most $\alpha_n |A|/2^l$ sets in $\mathcal{A}_n$ of cardinality at most 2^l each. Given a list of A and a list of $\mathcal{A}_n$, we can find such a covering. Let B be one of the covering sets containing x. Then, x can be specified by $A, n, l, \mathcal{A}_n$ and the index i of B among the covering sets. We need also $O(\log k + \log \log i + \log \log l + \log \log n)$ extra bits to separate the descriptions of A and $\mathcal{A}_n$, and the binary representations of i, n, l, from one another. Without loss of generality we can assume that k is less than n. Thus all the extra information and separator bits are included in $O(\log n)$ bits. Altogether, $C(B) \leq C(A) + m - l + O(\log n) \leq k + m - l + O(\log n)$, which shows that $g_x(l) \leq k + m - l + O(\log n) = g_x(m) + m - l + O(\log n)$. ■

Example 4: Lemma 1 shows that

$$C(x) - i - O(\log n) \leq g_x(i) \leq n - i + O(\log n),$$

for every $0 \leq i \leq n$. The right-hand inequality is obtained by setting $m = n$, $l = i$ in the lemma, yielding

$$g_x(i) = g_x(i) - g_x(n) + O(\log n) \leq n - i + O(\log n).$$

The left-hand inequality is obtained by setting $l = 0$, $m = i$ in the lemma, yielding

$$C(x) - g_x(i) = g_x(0) - g_x(i) + O(1) \leq i - 0 + O(\log n).$$

The last displayed equation can also be shown by a simple direct argument: x can be described by the minimal description of the set $A \in \mathcal{A}$ witnessing $g_x(i)$ and by the ordinal number of x in A. ◇

The rate-distortion function r_x differs from g_x by just a change of scale depending on the distortion family involved, provided certain computational requirements are fulfilled. See Appendix B for computability notions.

Lemma 2: Let $\mathcal{X} = \{0,1\}^*$, $\mathcal{Y}$, and d, be the source alphabet, destination alphabet, and distortion measure, respectively. Assume that the set $\{\langle x, y, \delta \rangle \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Q} : d(x,y) \leq \delta\}$ is decidable; that $\mathcal{Y}$ is recursively enumerable; and that for every n the cardinality of every ball in $\mathcal{A}_n^{d,\mathcal{Y}}$ of radius δ is at most $b_n(\delta)$ and at least $b_n(\delta)/\beta(n)$, where $\beta(n)$ is polynomial in n and $b_n(\delta)$ is a function of n, δ; and that the distortion family $\mathcal{A}^{d,\mathcal{Y}}$ satisfies properties 1 through 4. Then, for every $x \in \{0,1\}^n$ and every rational δ we have

$$r_x(\delta) = g_x(\lceil \log b_n(\delta) \rceil) + O(C(\delta) + \log n). \tag{4}$$

Proof: Fix n and a string x of length n. Consider the auxiliary function

$$\tilde{r}_x(\delta) = \min_{y \in \mathcal{Y}} \{C(B(y,\delta)) : d(x,y) \leq \delta\}. \tag{5}$$

We claim that $\tilde{r}_x(\delta) = r_x(\delta) + O(C(\delta) + \log n)$. Indeed, let y witness $r_x(\delta) = k$. Given y, δ, n we can compute a list of elements of the ball $B(y,\delta)$: for all strings x' of length n determine whether $d(x', y) \leq \delta$. Thus $C(B(y,\delta)) < k + O(C(\delta) + \log n)$, hence $\tilde{r}_x(\delta) < k + O(C(\delta) + \log n)$. Conversely, let $B(y,\delta)$ witness $\tilde{r}_x(\delta) = k$. Given a list of the elements of $B(y,\delta)$ and δ we can recursively enumerate $\mathcal{Y}$ to find the first element y' with $B(y',\delta) = B(y,\delta)$ (for every enumerated y' compute the list $B(y',\delta)$ and compare it to the given list $B(y,\delta)$). Then, $C(y') \leq k + O(C(\delta))$ and $d(x,y') \leq \delta$. Hence $r_x(\delta) < k + O(C(\delta))$.

Thus, it suffices to show that

$$\tilde{r}_x(\delta) = g_x(\lceil \log b_n(\delta) \rceil) + O(\log n).$$

($g_x(\lceil \log b_n(\delta) \rceil) \leq \tilde{r}_x(\delta)$) Assume $\tilde{r}_x(\delta) = k$ is witnessed by a distortion ball $B(y,\delta)$. By our assumption, the cardinality of $B(y,\delta)$ is at most $b_n(\delta)$, and hence $g_x(\lceil \log b_n(\delta) \rceil) \leq k$.

($\tilde{r}_x(\delta) \leq g_x(\lceil \log b_n(\delta) \rceil) + O(\log n)$) By Lemma 1, $g_x(l)$ and $g_x(l - m)$ differ by at most $m + O(\log n)$. Therefore, it suffices to show that $\tilde{r}_x(\delta) \leq g_x(\lceil \log b_n(\delta) \rceil - m)$ for some $m = O(\log n)$. We claim that this happens for $m = \lceil \log \beta(n) \rceil + 1$. Indeed, let $g_x(\lceil \log b_n(\delta) \rceil - m) = k$ be witnessed by a distortion ball B. Then, $|B| \leq 2^{\lceil \log b_n(\delta) \rceil}/(2\beta(n)) < b_n(\delta)/\beta(n)$. This implies that the radius of B is less than δ and hence B witnesses $\tilde{r}_x(\delta) \leq k$. ■

Remark 5: When measuring distortion we usually do not need rational numbers with numerator or denominator more than $n = |x|$. Then, the term $O(C(\delta))$ in (4) is absorbed by the term $O(\log n)$. Thus, describing the family of g_x's we obtain an approximate description of all possible rate-distortion functions r_x for given destination alphabet and distortion measure, satisfying the computability conditions, by using the transformation (4). An example of an approximate rate-distortion curve r_x for some string x of length n for Hamming distortion is given in Fig. 1. ◇

Remark 6: The computability properties of the functions r_x, d_x, and g_x, as well as the relation between the destination word for a source word and the related distortion ball, is explained in Appendix B. ◇

We present an approximate description of the family of possible g_x's below. It turns out that the description does not depend on the particular distortion family $\mathcal{A}$ as long as properties 1 through 4 are satisfied.

Definition 7: Let G_n stand for the class of all functions $g : \{0, 1, \ldots, n\} \to \mathcal{N}$ such that $g(n) = 0$ and $g(l - 1) \in \{g(l), g(l) + 1\}$ for all $1 \leq l \leq n$.

In other words, a function g is in G_n iff it is nonincreasing and the function $g(i) + i$ is nondecreasing and $g(n) = 0$. The following result is a generalization to arbitrary distortion measures of [22, Theorem IV.4] dealing with h_x (equaling d_x in the particular case of the distortion family $\mathcal{L}$). There, the precision in Item (ii) for source words of length n is $O(\log n)$, rather than the $O(\sqrt{n \log n})$ we obtain for general distortion families.

Theorem 1: Let $\mathcal{A}$ be a distortion family satisfying properties 1 through 4.

i) For every n and every string x of length n, the function $g_x(l)$ is equal to $g(l) + O(\log n)$ for some function $g \in G_n$.

ii) Conversely, for every n and every function g in G_n, there is a string x of length n such that for every $l = 0, \ldots, n$, $g_x(l) = g(l) + O(\sqrt{n \log n})$.

Remark 7: For fixed $k \leq n$ the number of different integer functions $g \in G_n$ with $g(0) = k$ is $\binom{n}{k}$. For $k = \frac{1}{2}n$, this number is of order $2^n/\sqrt{n}$, and therefore far greater than the number of strings x of length n and Kolmogorov complexity $C(x) = k = \frac{1}{2}n$ which is at most $2^{n/2}$. This explains the fact that in Theorem 1, Item (ii), we cannot precisely match a string x of length n to every function $g \in G_n$, and therefore have to use approximate shapes. $\Diamond$

Example 5: By Theorem 1, Item (ii), for every $g \in G_n$ there is a string x of length n that has g for its canonical rate-distortion function g_x up to an additive $O(\sqrt{n \log n})$ term. By (3), (4), and Remark 5, in the case of Hamming distortion, we have

$$r_x(\delta) = g_x(nH(\delta)) + O(\log n),$$

for $0 \leq \delta \leq \frac{1}{2}$. Fig. 1 gives the graph of a particular function $r(\delta) = g(nH(\delta))$ with g defined as follows: $g(l) = n(1 + H(\frac{1}{6}) - H(\frac{1}{3})) - l$ for $0 \leq l \leq nH(\frac{1}{6})$, $g(l) = n(1 + H(\frac{1}{6}) - H(\frac{1}{3}))$ for $nH(\frac{1}{6}) < l \leq nH(\frac{1}{3})$, and $g(l) = n - l$ for $nH(\frac{1}{3}) < l \leq n$. In this way, $g \in G_n$. Thus, there is a string x of length n with its rate-distortion graph $r_x(\delta)$ in a strip of size $O(\sqrt{n \log n})$ around the graph of $r(\delta)$. Note that r_x is almost constant on the segment $[\frac{1}{6}; \frac{1}{3}]$. Allowing the distortion to increase on this interval, all the way from $\frac{1}{6}$ to $\frac{1}{3}$, so allowing $n/6$ incorrect extra bits, we still cannot significantly decrease the rate. This means that the distortion-rate function $d_x(r)$ of x drops from $\frac{1}{3}$ to $\frac{1}{6}$ near the point $r = n(1 - H(\frac{1}{3}))$, exhibiting a very unsmooth behavior. $\Diamond$

V. Characterization

Theorem 2 states that a destination word that codes a given source word and minimizes the algorithmic mutual information with the given source word gives no advantage in rate over a minimal Kolmogorov complexity destination word that codes the source word. This theorem can be compared with Shannon's theorem, Theorem 4 in Appendix A, about the expected rate-distortion curve of a random variable.

Theorem 2: Let $\mathcal{A}$ be a distortion family satisfying properties 2 and 3, and $\mathcal{A}(x) = \{A \in \mathcal{A} : x \in A\}$. For every n and string x of length n and every $B \in \mathcal{A}(x)$ there is an $A \in \mathcal{A}(x)$ with $\lceil \log |A| \rceil = \lceil \log |B| \rceil$ and $C(A) \leq I(x : B) + O(\log C(B) + \log n)$, where $I(x : B) = C(B) - C(B \,|\, x)$ stands for the algorithmic information in x about B.

For further information about $I(x : B)$ see Definition 11 in Appendix C. The proof of Shannon's theorem, Theorem 4, and the proof of the current theorem are very different. The latter proof uses techniques that may be of independent interest. In particular, we use an online set cover algorithm where the sets come sequentially and we always have to have the elements covered that occur in a certain number of sets, Lemma 6 in Appendix F.

Example 6: Theorem 2 states that for an appropriate distortion family $\mathcal{A}$ of nonempty finite subsets of $\{0,1\}^*$ and for every string $x \in \{0,1\}^*$, if there exists an $A \in \mathcal{A}$ of cardinality 2^l or less containing x that has small algorithmic information about x, then there exists another set $B \in \mathcal{A}$ containing x that has also at most 2^l elements and has small Kolmogorov complexity itself.

For example, in the case of Hamming distortion, if for a given string x there exists a string y at Hamming distance δ from x that has small information about x, then there exists another string z that is also within distance δ of x and has small Kolmogorov complexity itself (not only small algorithmic information about x). Here x, y, z are strings of length n.

To see this, note that the Hamming distortion family of Example 2 satisfies properties 2 and 3. Consider Hamming balls in $\{0,1\}^n$. Let A be the ball of radius δ with center y. By Theorem 2 there is a Hamming ball $B \ni x$ with $\lceil \log |B| \rceil = \lceil \log |A| \rceil$ and $C(B) \leq C(A) - C(A \,|\, x) + O(\log C(A) + \log n)$. Notice that $\log C(A) = O(\log n)$ and thus we can drop $\log C(A)$ in this inequality. Without loss of generality we may assume that the radius of B at most $\delta < 1/2$ (otherwise we can cover B by a polynomial number of balls of radius δ by Lemma 5 and then replace B by a covering ball which contains x). Let z be the center of B. Thus z is at distance at most δ from x and it suffices to prove that $C(z) \leq I(x : y) + O(\log n)$. The information in the ball A equals that in the pair (y, δ). Therefore

$$\begin{aligned} C(z) &\leq C(B) \leq C(A) - C(A \,|\, x) + O(\log n) \\ &= C(y, \delta) - C((y, \delta) \,|\, x) + O(\log n). \end{aligned}$$

Since $C(\delta) \leq \log n + O(1)$, we obtain

$$\begin{aligned} C(z) &\leq C(y) - C(y \,|\, x) + O(\log n) \\ &= I(x : y) + O(\log n). \end{aligned}$$

$\Diamond$

VI. Fitness of Destination Word

In Theorem 3 we show that if a destination word of a certain maximal Kolmogorov complexity has minimal distortion with respect to the source word, then it also is the (almost) best-fitting destination word in the sense (explained below) that among all destination words of that Kolmogorov complexity it has the most properties in common with the source word. "Fitness" of individual strings to an individual destination word is hard, if not impossible, to describe in the probabilistic framework. However, for the combinatoric and computational notion of Kolmogorov complexity it is natural to describe this notion using "randomness deficiency" as in Definition 8.

Reference [22] uses 'fitness' with respect to the particular distortion family $\mathcal{L}$. We briefly overview the generalization to arbitrary distortion families satisfying properties 2 and 3 (details, formal statements and proofs about $\mathcal{L}$ can be found in the cited reference). The goodness of fit of a destination word y for a source word x with respect to an arbitrary distortion family $\mathcal{A}$ is defined by the randomness deficiency of x in the distortion ball $B(y, \delta)$ with $\delta = d(x, y)$. The lower the randomness deficiency, the better is the fit.

Definition 8: The *randomness deficiency* of x in a set A with $x \in A$ is defined as $\delta(x \mid A) = \log |A| - C(x \mid A)$. If $\delta(x \mid A)$ is small then x is a *typical* element of A. Here "small" is taken as $O(1)$ or $O(\log n)$ where $n = |x|$, depending on the context of the future statements.

The randomness deficiency can be little smaller than 0, but not more than a constant.

Definition 9: Let $\beta > 1$ be an integer parameter and $P \subseteq A$. We say P is a *property* in A if P is a 'majority' subset of A, that is, $|P| \geq (1 - 2^{-\beta})|A|$. We say that $x \in A$ *satisfies* property P if $x \in P$.

If the randomness deficiency $\delta(x \mid A)$ is not much greater than 0, then there are no simple special properties that single x out from the majority of strings to be drawn from A. This is not just terminology: If $\delta(x|A)$ is small enough, then x satisfies *all* properties of low Kolmogorov complexity in A (Lemma 4 in Appendix D). If A is a set containing x such that $\delta(x \mid A)$ is small then we say that A is a set of good fit for x. In [22] the notion of models for x is considered: Every finite set of strings containing x is a *model* for x. Let x be a string of length n and choose an integer i between 0 and n. Consider models for x of Kolmogorov complexity at most i. In [22, Theorem IV.8 and Remark IV.10] show for the distortion family $\mathcal{L}$ that x has minimal randomness deficiency in every set that witnesses $h_x(i)$ (for $\mathcal{L}$ we have $h_x(i) = d_x(i)$), ignoring additive $O(\log n)$ terms. That is, up to the stated precision every such witness set is the best-fitting model that is possible at model Kolmogorov complexity at most i. It is remarkable, and in fact unexpected to the authors, that the analogous result holds for arbitrary distortion families provided they satisfy properties 2 and 3.

Theorem 3: Let $\mathcal{A}$ be a distortion family satisfying properties 2 and 3 and x a string of length n. Let B be a set in $\mathcal{A}$ with $x \in B$. Let A_x be a set of minimal Kolmogorov complexity among the sets $A \in \mathcal{A}$ with $x \in A$ and $\lceil \log |A| \rceil = \lceil \log |B| \rceil$. Then

$$C(A_x) + \log |A_x| - C(x) \leq \delta(x \mid B) + O(\log C(B) + \log n).$$

Lemma 3: For every set A with $x \in A$

$$C(A) + \log |A| - C(x) \geq \delta(x \mid A) \tag{6}$$

up to a $O(\log n)$ additive term.

Proof: Inequality (6) means that

$$C(A) + \log |A| - C(x) \geq \log |A| - C(x \mid A) + O(\log n)$$

that is, $C(x) \leq C(A) + C(x \mid A) + O(\log n)$. The latter inequality follows from the general inequality $C(x) \leq C(x, y) \leq C(y) + C(x \mid y) + O(\log C(x \mid y))$, where $C(x \mid y) \leq C(x) + O(1) \leq n + O(1)$. ■

A set A with $x \in A$ is an algorithmic *sufficient statistic* for x if $C(A) + \log |A|$ is close to $C(x)$. Lemma 3 shows that every sufficient statistic for x is a model of a good fit for x.

Example 7: Consider the elements of every $A \in \mathcal{A}$ uniformly distributed. Assume that we are given a string x that was obtained by a random sampling from an unknown set $B \in \mathcal{A}$ satisfying $C(B) \leq n = |x|$. Given x we want to recover B, or some $A \in \mathcal{A}$ that is "a good hypothesis to be the source of x" in the sense that the randomness deficiency $\delta(x \mid A)$ is small. Consider the set A_x from Theorem 3 as such a hypothesis. We claim that with high probability $\delta(x \mid A_x)$ is of order $O(\log n)$. More specifically, for every β the probability of the event $\delta(x \mid A_x) > \beta$ is less than $2^{-\beta + O(\log n)}$, which is negligible for $\beta = O(\log n)$. Indeed, if x is chosen uniformly at random in B, then with high probability (Appendix D) the randomness deficiency $\delta(x \mid B)$ is small. That is, with probability more than $1 - 2^{-\beta}$ we have $\delta(x \mid B) \leq \beta$. By Theorem 3 and (6) we also have $\delta(x \mid A_x) \leq \delta(x \mid B) + O(\log n)$. Therefore, the probability of the event $\delta(x \mid A_x) > \beta$ is less than $2^{-\beta + O(\log n)}$. ◊

Example 8: Theorem 3 says that for fixed log-cardinality l the model that has minimal Kolmogorov complexity has also minimal randomness deficiency among models of that log-cardinality. Since g_x satisfies Lemma 1, we have also that for every k the model of Kolmogorov complexity at most k that minimizes the log-cardinality also minimizes randomness deficiency among models of that Kolmogorov complexity. These models can be computed in the limit, in the first case by running all programs up to k bits and always keeping the one that outputs the smallest set in $\mathcal{A}$ containing x, and in the second case by running all programs up to $n = |x|$ bits and always keeping the shortest one that outputs a set in $\mathcal{A}$ containing x having log-cardinality at most l. ◊

VII. Denoising

In Theorem 3, using (6) we obtain

$$\delta(x \mid A_x) \leq \delta(x \mid B) + O(\log C(B) + \log n). \tag{7}$$

This gives a method to identify good-fitting models for x using compression, as follows. Let $k = C(A_x)$ and $l = \lceil \log |B| \rceil$. If A_x is a set of minimal Kolmogorov complexity among sets $A \in \mathcal{A}$ with $x \in A$ and $\lceil \log |A| \rceil = l$, then by (7) the hypothesis "x is chosen at random in A_x" is (almost) at least as plausible as the hypothesis "x is chosen at random in B" for every simply described $B \in \mathcal{A}$ (say, $\log C(B) = O(\log n)$) with $\lceil \log |B| \rceil = l$.

Let us look at an example of denoising by compression (in the ideal sense of Kolmogorov complexity) for Hamming distortion. Fix a target string y of length n and a distortion $0 \leq \delta \leq \frac{1}{2}$. (This string y functions as the destination word.) Let a string x be a noisy version of y by changing at most $n\delta$ randomly chosen bits in y (string x functions as the source word). That is, the string x is chosen uniformly at random in the Hamming ball $B = B(y, \delta)$. Let $\hat{x}$ be a string witnessing $r_x(\delta)$, that is, $\hat{x}$ is a string of minimal Kolmogorov complexity with $d(x, \hat{x}) \leq \delta$ and $r_x(\delta) = C(\hat{x})$. We claim that at distortion δ the string $\hat{x}$ is a good candidate for a denoised version of x, that is, the target string y. This means that in the two-part description $(\hat{x}, \hat{x} \oplus x)$ of x, the second part (the bitwise XOR of x and $\hat{x}$) is noise: $\hat{x} \oplus x$ is a random string in the Hamming ball $B(00\ldots0, \delta)$ in the sense that $\delta(\hat{x} \oplus x \mid B(00\ldots0, \delta))$ is negligible. Moreover, even the conditional Kolmogorov complexity $C(\hat{x} \oplus x \mid \hat{x})$ is close to $\log b(\delta)$.

Indeed, let $l = \lceil \log |B| \rceil$. By Definition 5 of g_x, Theorem 3 implies that

$$g_x(l) + l - C(x) \leq \delta(x \mid B),$$

ignoring additive terms of $O(\log n)$ and observing that the additive term $\log C(B)$ is absorbed by $O(\log n)$. For every x, the rate-distortion function r_x of x differs from g_x just by changing the scale of the argument as in (4). More specifically, we have $r_x(\delta) = g_x(l)$ and hence

$$r_x(\delta) + l - C(x) \leq \delta(x \mid B).$$

Since we assume that x is chosen uniformly at random in B, the randomness deficiency $\delta(x \mid B)$ is small, say $O(\log n)$ with high probability. Since $r_x(\delta) = C(\hat{x}) = C(B(\hat{x}, \delta)) + O(C(\delta))$, $C(\delta) = O(\log n)$, and $l = \lceil \log b(\delta) \rceil$, it follows that with high probability, and the equalities up to an additive $O(\log n)$ term,

$$0 = C(\hat{x}) + l - C(x) = C(B(\hat{x}, \delta)) + \log b(\delta) - C(x).$$

Since by construction $x \in B(\hat{x}, \delta)$, the displayed equation shows that the ball $B(\hat{x}, \delta)$ is a sufficient statistic for x. This implies that x is a typical element of $B(\hat{x}, \delta)$, that is, $C(x \oplus \hat{x} \mid \hat{x}) = C(x \mid \hat{x}) = C(x \mid B(\hat{x}, \delta), p)$ is close to $\log b(\delta)$. Here p is an appropriate program of $O(\mathcal{C}(\delta)) = O(\log n)$ bits.

This provides a method of denoising via compression, at least in theory. In order to use the method practically, admittedly with a leap of faith, we ignore the ubiquitous $O(\log n)$ additive terms, and use real compressors to approximate the Kolmogorov complexity, similar to what was done in [10], [11]. The Kolmogorov complexity is not computable and can be approximated by a computable process from above but not from below, while a real compressor is computable. Therefore, the approximation of the Kolmogorov complexity by a real compressor involves for some arguments errors that can be high and are in principle unknowable. Despite all these caveats it turns out that the practical analogue of the theoretical method works surprisingly well in all experiments we tried [15].

As an example, we approximated the distortion-rate function of a noiseless cross called the *target*. It consists of a monochrome image of 1188 black pixels together with 2908 surrounding white pixels, forming a plane of 64 × 64 black-or-white pixels. Added are 377 pixels of artificial noise inverting 109 black pixels and 268 white pixels. This way we obtain a noisy cross called the *input*. The input is in effect a pixelwise exclusive OR of the target and noise. The distortion used is Hamming distortion. At every rate r $(0 \leq r \leq 3500)$ we compute a set $M(r)$ of candidates. Every candidate consists of the 64 × 64 pixel plane divided into black pixels and white pixels. Every candidate approximates the input in a certain sense and a compressed version requires at most r bits. For every (uncompressed) candidate in $M(r)$ the distortion to the input is computed. The candidate in $M(r)$ that minimizes the distortion is called the "best" candidate at rate r.

Fig. 2 shows two graphs. The first graph hits the horizontal axis at about 3178 bits. On the horizontal axis it gives the rate, and on the vertical axis it denotes the distortion to the input of the best candidate at every rate. The line hits zero distortion at rate about 3178, when the input is retrieved as the best candidate (attached to this point). The second graph hits the horizontal axis at about 260 bits. The horizontal axis denotes again the rate, but now the vertical axis denotes the distortion between the best candidate and the target. The line hits almost zero distortion (three bits flipped) at rate about 260. There an image that is almost the target is retrieved as the best candidate (attached to this point). The three wrong bits are two at the bottom left corner and one in the upper right armpit. The hitting of the horizontal axis by the second graph coincides with a sharp slowing of the rate of decrease of the first graph. Subsequently, the second graph rises again because the best candidate at that rate starts to model more of the noise present in the input. Thus, the second graph shows us the denoising of the input, underfitting left of the point of contact with the horizontal axis, and overfitting right of that point. This point of best denoising can also be deduced from the first graph, where it is the point where the distortion-rate curve sharply levels off. Since this point has distortion of only 3 to the target, the distortion-rate function separates structure and noise very well in this example.

In the experiments in [15], a specially written block sorting compression algorithm with a move-to-front scheme as described in [3] was used. The algorithm is very similar to a number of common general purpose compressors, such as bzip2 and zzip, but it is simpler and faster for small inputs; the source code (in C) is available from the authors of [15].

Appendix

A. Shannon Rate Distortion

Classical rate-distortion theory was initiated by Shannon in [17], [18], and we briefly recall his approach. Let $\mathbf{X}$ and $\mathbf{Y}$ be finite alphabets. A single-letter distortion measure is a function d that maps elements of $\mathbf{X} \times \mathbf{Y}$ to the reals. Define the distortion between words x and y of the same length n over alphabets $\mathbf{X}$ and $\mathbf{Y}$, respectively, by extending d:

$$d(x, y) = \frac{1}{n} \sum_{i=1}^{n} d(x_i, y_i).$$

Let X be a random variable with values in $\mathbf{X}$. Consider the random variable X^n with values in $\mathbf{X}^n$, that is, the sequence $X_1, \ldots, X_n$ of n independent copies of X. We want to encode words in $\mathbf{X}^n$ by code words over $\mathbf{Y}^n$ so that the number of all code words is small and the expected distortion between outcomes of X^n and their codes is small. The tradeoff between the expected distortion and the number of code words used is expressed by the Shannon *rate-distortion* function r^n. This function maps every $\delta \in \mathcal{R}$ to the minimal natural number r (we call r the *rate*) having the following property: There is an encoding function $E : \mathbf{X}^n \to \mathbf{Y}^n$ with a range of cardinality at most 2^r such that the expected distortion between the outcomes of X^n and their corresponding codes is at most δ. That is

$$r^n(\delta) = \min_{E} \{ \lceil \log |E(\mathbf{X}^n)| \rceil : \mathbf{E} d(x, E(x)) \leq \delta \} \tag{8}$$

the expectation $\mathbf{E}$ taken over the probabilities of the x's in $\mathbf{X}^n$.

In [18] Shannon gave the following nonconstructive asymptotic characterization of $r^n(\delta)$. Let Z be a random variable with values in $\mathbf{Y}$. Let $H(Z)$, $H(Z \mid X)$ stand for the Shannon

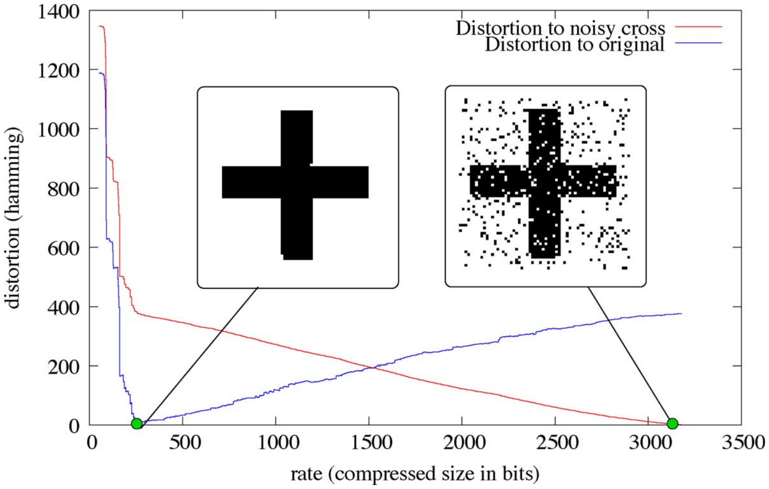

Fig. 2. Denoising of the noisy cross.

entropy and conditional Shannon entropy, respectively. Let $I(X;Z) = H(Z) - H(Z\,|\,X)$ denote the mutual information in X and Z, and $\mathbf{E}d(X,Z)$ stand for the expected value of $d(x,z)$ with respect to the joint probability $P(X=x, Z=z)$ of the random variables X and Z. For a real δ, let $R(\delta)$ denote the minimal $I(X;Z)$ subject to $\mathbf{E}d(X,Z) \leq \delta$. That such a minimum is attained for all δ can be shown by compactness arguments.

Theorem 4: For every n and δ we have $r^n(\delta) \geq nR(\delta)$. Conversely, for every δ and every positive ϵ, we have $r^n(\delta+\epsilon) \leq n(R(\delta)+\epsilon)$ for all large enough n.

B. Computability

In 1936, Turing [21] defined the hypothetical "Turing machine" whose computations are intended to give an operational and formal definition of the intuitive notion of computability in the discrete domain. These Turing machines compute integer functions, the *computable* functions. By using pairs of integers for the arguments and values we can extend computable functions to functions with rational arguments and/or values. The notion of computability can be further extended, see, for example, [9]: A function f with rational arguments and real values is *upper semicomputable* if there is a computable function $\phi(x,k)$ with x an rational number and k a nonnegative integer such that $\phi(x,k+1) \leq \phi(x,k)$ for every k and $\lim_{k\to\infty} \phi(x,k) = f(x)$. This means that f can be computably approximated from above. A function f is *lower semicomputable* if $-f$ is upper semicomputable. A function is called *semicomputable* if it is either upper semicomputable or lower semicomputable or both. If a function f is both upper semicomputable and lower semicomputable, then f is *computable*. A countable set S is *computably (or recursively) enumerable* if there is a Turing machine T that outputs all and only the elements of S in some order and does not halt. A countable set S is *decidable (or recursive)* if there is a Turing machine T that decides for every candidate a whether $a \in S$ and halts.

Example 9: An example of a computable function is $f(n)$ defined as the nth prime number; an example of a function that is upper semicomputable but not computable is the Kolmogorov complexity function C in Appendix C. An example of a recursive set is the set of prime numbers; an example of a recursively enumerable set that is not recursive is $\{x \in \mathcal{N} : C(x) < |x|\}$.◊

Let $\mathcal{X} = \{0,1\}^*$, and $\mathcal{Y}$ and the distortion measure d be given. Assume that $\mathcal{Y}$ is recursively (= computably) enumerable and the set $\{\langle x,y,\delta\rangle \in \mathcal{X}\times\mathcal{Y}\times\mathcal{Q} : d(x,y) \leq \delta\}$ is decidable. Then r_x is upper semicomputable. Namely, to determine $r_x(\delta)$ proceed as follows. Recall that U is the reference universal Turing machine. Run $U(p)$ for all p dovetailed fashion (in stage k of the overall computation execute the ith computation step of the $(k-i)$th program). Interleave this computation with a process that recursively enumerates $\mathcal{Y}$. Put all enumerated elements of $\mathcal{Y}$ in a set $\mathcal{W}$. Whenever $U(p)$ halts we put the output in a set $\mathcal{U}$. After every step in the overall computation we determine the minimum length of a program p such that $U(p) \in \mathcal{W} \bigcap \mathcal{U}$ and $d(x, U(p)) \leq \delta$. We call p a *candidate* program. The minimal length of all candidate programs can only decrease in time and eventually becomes equal to $r_x(\delta)$. Thus, this process upper semicomputes $r_x(\delta)$.

The function g_x is also upper semicomputable. The proof is similar to that used to prove the upper semicomputability of r_x. It follows from [22] that in general d_x, and, hence, its "inverse" r_x and by Lemma 2 the function g_x, are not computable.

Assume that the set $\mathcal{Y}$ is recursively enumerable and the set $\{\langle x,y,\delta\rangle \in \mathcal{X}\times\mathcal{Y}\times\mathcal{Q} : d(x,y) \leq \delta\}$ is decidable. Assume that the resulting distortion family $\mathcal{A}^{d,\mathcal{Y}}$ satisfies Property 2. There is a relation between destination words and distortion balls. This relation is as follows.

(i) Communicating a destination word y for a source word x knowing a rational upper bound δ for the distortion $d(x,y)$ involved is the same as communicating a distortion ball of radius δ containing x.

(ii) Given (a list of the elements of) a distortion ball B we can upper semicompute the least distortion δ such that $B = B(y,\delta)$ for some $y \in \mathcal{Y}$.

Ad (i). This implies that the function $\tilde{r}_x(\delta)$ defined in (5) differs from $r_x(\delta)$ by $O(C(\delta)+\log|x|)$. See the proof of Lemma 2.

Ad (ii). Let B be a given ball. Recursively enumerating $\mathcal{Y}$ and the possible $\beta \in \mathcal{Q}$, we find for every newly enumerated element of $y \in \mathcal{Y}$ whether $B(y, \beta) = B$ (see the proof of Lemma 2 for an algortihm to find a list of elements of $B(y, \beta)$ given y, β). Put these β's in a set $\mathcal{W}$. Consider the least element of $\mathcal{W}$ at every computation step. This process upper semicomputes the least distortion δ corresponding to the distortion ball B.

C. *Kolmogorov Complexity*

For precise definitions, notation, and results see the text [9]. Informally, the Kolmogorov complexity, or algorithmic entropy, $C(x)$ of a string x is the length (number of bits) of a shortest binary program (string) to compute x on a fixed reference universal computer (such as a particular universal Turing machine). Intuitively, $C(x)$ represents the minimal amount of information required to generate x by any effective process. The conditional Kolmogorov complexity $C(x \,|\, y)$ of x relative to y is defined similarly as the length of a shortest binary program to compute x, if y is furnished as an auxiliary input to the computation.

Let $T_1, T_2, \ldots$ be a standard enumeration of all (and only) Turing machines with a binary input tape, for example the lexicographic length-increasing ordered syntactic Turing machine descriptions, [9], and let $\phi_1, \phi_2, \ldots$ be the enumeration of corresponding functions that are computed by the respective Turing machines (T_i computes ϕ_i). These functions are the *computable (or recursive)* functions. For the development of the theory we actually require the Turing machines to use *auxiliary* (also called *conditional*) information, by equipping the machines with a special read-only auxiliary tape containing this information at the outset. Let $\langle \cdot, \cdot \rangle$ be a computable one to one *pairing function* on the natural numbers (equivalently, strings) mapping $\{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ with $|\langle u, v \rangle| \leq |u| + |v| + O(\log(|u|))$. (We need the extra $O(\log(|u|))$ bits to separate u from v. For Kolmogorov complexity, it is essential that there exists a pairing function such that the length of $\langle u, v \rangle$ is equal to the sum of the lengths of u, v plus a small value depending only on $|u|$.) We denote the function computed by a Turing machine T_i with p as input and y as conditional information by $\phi_i(p, y)$.

One of the main achievements of the theory of computation is that the enumeration $T_1, T_2, \ldots$ contains a machine, say T_u, that is computationally universal in that it can simulate the computation of every machine in the enumeration when provided with its index. It does so by computing a function ϕ_u such that $\phi_u(\langle i, p \rangle, y) = \phi_i(p, y)$ for all i, p, y. We fix one such machine and designate it as the *reference universal Turing machine* or *reference Turing machine* for short.

Definition 10: The *conditional Kolmogorov complexity* of x given y (as auxiliary information) *with respect to Turing machine* T_i is

$$C_i(x \,|\, y) = \min_p \{|p| : \phi_i(p, y) = x\}. \tag{9}$$

The *conditional Kolmogorov complexity* $C(x \,|\, y)$ is defined as the conditional Kolmogorov complexity $C_u(x \,|\, y)$ with respect to the reference Turing machine T_u usually denoted by U. The *unconditional* version is set to $C(x) = C(x \,|\, \epsilon)$.

Kolmogorov complexity $C(x \,|\, y)$ has the following crucial property: $C(x \,|\, y) \leq C_i(x \,|\, y) + c_i$ for all i, x, y, where c_i depends only on i (asymptotically, the reference Turing machine is not worse than any other machine). Intuitively, $C(x \,|\, y)$ represents the minimal amount of information required to generate x by any effective process from input y. The functions $C(\cdot)$ and $C(\cdot \,|\, \cdot)$, though defined in terms of a particular machine model, are machine-independent up to an additive constant and acquire an asymptotically universal and absolute character through Church's thesis, see for example [9], and from the ability of universal machines to simulate one another and execute any effective process. The Kolmogorov complexity of an individual finite object was introduced by Kolmogorov [7] as an absolute and objective quantification of the amount of information in it. The information theory of Shannon [17], on the other hand, deals with *average* information *to communicate* objects produced by a *random source*. Since the former theory is much more precise, it is surprising that analogs of theorems in information theory hold for Kolmogorov complexity, be it in somewhat weaker form. For example, let X and Y be random variables with a joint distribution. Then, $H(X, Y) \leq H(X) + H(Y)$, where $H(X)$ is the entropy of the marginal distribution of X. Similarly, let $C(x, y)$ denote $C(\langle x, y \rangle)$ where $\langle \cdot, \cdot \rangle$ is a standard pairing function as defined previously and x, y are strings. Then we have $C(x, y) \leq C(x) + C(y) + O(\log C(x))$. Indeed, there is a Turing machine T_i that provided with $\langle p, q \rangle$ as an input computes $\langle U(p), U(q) \rangle$ (where U is the reference Turing machine). By construction of T_i, we have $C_i(x, y) \leq C(x) + C(y) + O(\log C(x))$, hence $C(x, y) \leq C(x) + C(y) + O(\log C(x))$.

Another interesting similarity is the following: $I(X; Y) = H(Y) - H(Y \,|\, X)$ is the (probabilistic) *information in random variable X about random variable Y*. Here $H(Y \,|\, X)$ is the conditional entropy of Y given X. Since $I(X; Y) = I(Y; X)$ we call this symmetric quantity the *mutual (probabilistic) information.*

Definition 11: The *(algorithmic) information in x about y* is $I(x : y) = C(y) - C(y \,|\, x)$, where x, y are finite objects like finite strings or finite sets of finite strings.

It is remarkable that also the algorithmic information in one finite object about another one is symmetric: $I(x : y) = I(y : x)$ up to an additive term logarithmic in $C(x) + C(y)$. This follows immediately from the *symmetry of information* property due to Kolmogorov and Levin

$$\begin{aligned} C(x, y) &= C(x) + C(y \,|\, x) + O(\log(C(x) + C(y))) \\ &= C(y) + C(x \,|\, y) + O(\log(C(x) + C(y))). \end{aligned} \tag{10}$$

D. *Randomness Deficiency and Fitness*

Randomness deficiency of an element x of a finite set A according to Definition 8 is related with the fitness of $x \in A$ (identified with the fitness of set A as a model for x) in the sense of x having most properties represented by the set A. Properties are identified with large subsets of A whose Kolmogorov complexity is small (the 'simple' subsets).

Lemma 4: Let β, γ be constants. Assume that P is a subset of A with $|P| \geq (1 - 2^{-\beta})|A|$ and $C(P \,|\, A) \leq \gamma$. Then the randomness deficiency $\delta(x \mid A)$ of every $x \in A \setminus P$ satisfies $\delta(x \mid A) > \beta - \gamma - O(\log\log|A|)$

Proof: Since $\delta(x \mid A) = \log|A| - C(x \mid A)$ and $C(x \mid A) \leq C(x \mid A, P) + C(P \mid A) + O(\log C(x \mid A, P))$, while $C(x \mid A, P) \leq -\beta + \log|A| + O(1) \leq \log|A| + O(1)$, we obtain $\delta(x \mid A) > \beta - \gamma - O(\log\log|A|)$. ■

The randomness deficiency measures our disbelief that x can be obtained by random sampling in A (where all elements of A are equiprobable). For every A, the randomness deficiency of almost all elements of A is small: The number of $x \in A$ with $\delta(x \mid A) > \beta$ is fewer than $|A|2^{-\beta}$. This can be seen as follows. The inequality $\delta(x \mid A) > \beta$ implies $C(x \mid A) < \log|A| - \beta$. Since $1 + 2 + 2^2 + \cdots + 2^{i-1} = 2^i - 1$, there are less than $2^{\log|A|-\beta}$ programs of fewer than $\log|A| - \beta$ bits. Therefore, the number of x's satisfying the inequality $C(x \mid A) < \log|A| - \beta$ cannot be larger. Thus, with high probability the randomness deficiency of an element randomly chosen in A is small. On the other hand, if $\delta(x \mid A)$ is small, then there is no way to refute the hypothesis that x was obtained by random sampling from A: Every such refutation is based on a simply described property possessed by a majority of elements of A but not by x. Here it is important that we consider only simply described properties, since otherwise we can refute the hypothesis by exhibiting the property $P = A \setminus \{x\}$.

E. Covering Coefficient for Hamming Distortion

The authors find it difficult to believe that the covering result in the lemma below is new. But neither a literature search nor the consulting of experts has turned up an appropriate reference.

Lemma 5: Consider the distortion family $\mathcal{H}_n$. For all $0 \leq d \leq \delta \leq \frac{1}{2}$ every Hamming ball of radius δ in $\mathcal{H}_n$ can be covered by at most $\alpha_n b(\delta)/b(d)$ Hamming balls of radius d in $\mathcal{H}_n$, where α_n is a polynomial in n.

Proof: Fix a ball with center y and radius $\delta = j/n \leq \frac{1}{2}$ where j is a natural number. All the strings in the ball that are at Hamming distance at most d from y can be covered by one ball of radius d with center y. Thus it suffices, for every Δ of the form i/n with $i = 2, 3, \ldots, j$ (such that $d < \Delta \leq \delta$), to cover the set of all the strings at distance precisely Δ from y by $n^{c+1}b(\delta)/b(d)$ balls of radius d for some fixed constant c. Then the ball $B(y, \delta)$ is covered by at most $jn^{c+1}b(\delta)/b(d) \leq n^{c+2}b(\delta)/b(d)$ balls of radius d.

Fix Δ and let the Hamming sphere S denote the set of all strings at distance precisely Δ from y. Let f be the solution to the equation $d + f(1-2d) = \Delta$ rounded to the closest rational of the form i/n. Since $d < \Delta \leq \delta \leq \frac{1}{2}$ this equation has a unique solution and it lies in the closed real interval $[0, 1]$. Consider a ball B of radius d with a random center z at distance f from y. Assume that all centers at distance f from y are chosen with equal probabilities $1/s(f)$ where $s(f)$ is the number of points in a Hamming sphere of radius f.

Claim 1: Let x be a particular string in S. Then

$$\Pr(x \in B) \geq \frac{b(d)}{n^c b(\delta)}$$

for some fixed positive constant c.

Proof: Fix a string z at distance f from y. We first claim that the ball B of radius d with center z covers $b(d)/n^c$ strings in S. Without loss of generality, assume that the string y consists of only zeros and string z consists of fn ones and $(1-f)n$ zeros. Flip a set of fdn ones and a set of $(1-f)dn$ zeros in z to obtain a string u. The total number of flipped bits is equal to dn and therefore u is at distance d from z. The number of ones in u is $fn - fdn + (1-f)dn = \Delta n$ and therefore $u \in S$. Different choices of the positions of the same numbers of flipped bits result in different strings in S. The number of ways to choose the flipped bits is equal to

$$\binom{fn}{fdn}\binom{(1-f)n}{(1-f)dn}.$$

By Stirling's formula, this is at least

$$2^{fnH(d)+(1-f)nH(d)-O(\log n)} = 2^{nH(d)-O(\log n)} \geq \frac{b(d)}{n^c},$$

where the last inequality follows from (3). Therefore a ball B as above covers at least $b(d)/n^c$ strings of S. The probability that a ball B, chosen uniformly at random as above, covers a particular string $x \in S$ is the same for every such x since they are in symmetric position. The number of elements in a Hamming sphere is smaller than the cardinality of a Hamming ball of the same radius, $|S| \leq b(\delta)$. Hence with probability

$$\frac{b(d)}{n^c|S|} \geq \frac{b(d)}{n^c b(\delta)}$$

a random ball B covers a particular string x in S. ■

By Claim 1, the probability that a random ball B does not cover a particular string $x \in S$ is at most $1 - b(d)/(n^c b(\delta))$. The probability that no ball out of N randomly drawn such balls B covers a particular $x \in S$ (all balls are equiprobable) is at most

$$\left(1 - \frac{b(d)}{n^c b(\delta)}\right)^N < e^{-Nb(d)/(n^c b(\delta))}.$$

For $N = n^{c+1}b(\delta)/b(d)$, the exponent of the right-hand side (RHS) of the last inequality is $-n$, and the probability that x is not covered is at most e^{-n}. This probability remains exponentially small even after multiplying by $|S| \leq 2^n$, the number of different x's in S. Hence, with probability at least $1 - (2/e)^n$ we have that N random balls of the given type cover all the strings in S. Therefore, there exists a deterministic selection of N such balls that covers all the strings in S. The lemma is proved. (A more accurate calculation shows that the lemma holds with $\alpha_n = O(n^4)$.) ■

Corollary 1: Since all strings of length n are either in the Hamming ball $B(00\ldots0, \frac{1}{2})$ or in the Hamming ball $B(11\ldots1, \frac{1}{2})$ in $\mathcal{H}_n$, the lemma implies that the set $\{0,1\}^n$ can be covered by at most

$$N = \frac{2\alpha_n 2^n}{b(d)}$$

balls of radius d for every $0 \leq d \leq \frac{1}{2}$. (A similar, but direct, calculation lets us replace the factor $2\alpha_n$ by n.)

F. Proofs of the Theorems

Proof: of Theorem 1. (i) Lemma 1 (assuming properties 1 through 4) implies that the canonical structure function g_x of every string x of length n is close to some function in the family G_n. This can be seen as follows. Fix x and construct g inductively for $n, n-1, \ldots, 0$. Define $g(n) = 0$ and

$$g(l-1) = \begin{cases} g(l) + 1 & \text{if } g(l) < g_x(l-1), \\ g(l) & \text{otherwise.} \end{cases}$$

By construction this function belongs to the family G_n. Let us show that $g_x(l) = g(l) + O(\log n)$. First, we prove that

$$g(l) \leq g_x(l) \tag{11}$$

by induction on $l = n, n-1, \ldots, 0$. For $l = n$ the inequality is straightforward, since by definition $g(n) = 0$. Let $0 \leq l \leq n$. Assume that $g(i) \leq g_x(i)$ for $i = n, n-1, \ldots, l$. If $g(l) < g_x(l-1)$ then $g(l-1) = g(l) + 1$ and therefore $g(l-1) \leq g_x(l-1)$. If $g(l) \geq g_x(l-1)$ then $g(l-1) = g(l) \geq g_x(l-1) \geq g_x(l) \geq g(l)$ and hence $g(l-1) = g_x(l-1)$.

Second, we prove that

$$g_x(l) \leq g(l) + O(\log n)$$

for every $l = 0, 1, \ldots, n$. Fix an l and consider the least m with $l \leq m \leq n$ such that $g_x(m) = g(m)$. If there is no such m we take $m = n$ and observe that $g_x(n) = O(\log n) = g(n) + O(\log n)$. This way, $g_x(m) = g(m) + O(\log n)$ and for every $l < l' \leq m$ we have $g(l'-1) < g_x(l'-1)$ due to inequality (11) and definition of m. Then $g_x(l'-1) > g(l'-1) \geq g(l')$, since we know that g is nonincreasing. Then, by the definition of g we have $g(l'-1) = g(l') + 1$. Thus, we have $g(l) = g(m) + m - l$. Hence, $g_x(l) \leq g_x(m) + m - l + O(\log n) = g(m) + m - l + O(\log n) = g(l) + O(\log n)$, where the inequality follows from Lemma 1, the first equality from the assumption that $g_x(m) = g(m) + O(\log n)$, and the second equality from the previous sentence.

(ii) In [22, Theorem IV.4] we proved a similar statement for the special distortion family $\mathcal{L}$ with an error term of $O(\log n)$. However, for the special case $\mathcal{L}$ we can let x be equal to the first x satisfying the inequality $g_x(l) \geq g(l) - O(\log n)$ for every l. In the general case this does not work any more. Here we construct x together with sets ensuring the inequalities $g_x(l) \leq g(l) + O(\sqrt{n \log n})$ for every $l = 0, \ldots, n$.

The construction is as follows. Divide the segment $\{0, 1, \ldots, n\}$ into $N = \sqrt{n / \log n}$ subsegments of length $\sqrt{n \log n}$ each. Let $l_0 = n > l_1 > \cdots > l_N = 0$ denote the end points of the resulting subsegments.

To find the desired x, we run the nonhalting algorithm below that takes n and $\mathcal{A}_n$ as input together with the values of the function g in the points $l_0, \ldots, l_N$. Let $\delta(n)$ be a computable integer valued function of n of the order $\sqrt{n \log n}$ that will be specified later.

Definition 12: Let $i = 0, 1, \ldots, N$. A set $F \in \mathcal{A}_n$ is called *i-forbidden* if $|F| \leq 2^{l_i}$ and $C(F) < g(l_i) - \delta(n)$. A set is called *forbidden* if it is i-forbidden for some $i = 0, 1, \ldots, N$.

We wish to find an x that is outside all forbidden sets (since this guarantees that $g_x(l_i) \geq g(l_i) - \delta(n)$ for every i). Since $C(\,\cdot\,)$ is upper semicomputable, moreover property 3 holds, and we are also given n and $g(l_0), \ldots, g(l_N)$, we are able to find all forbidden sets using the following subroutine.

Subroutine $(n, \mathcal{A}_n, g(l_0), g(l_1), \ldots, g(l_N))$:

for every $F \in \mathcal{A}_n$ upper semicompute $C(F)$; every time we find $C(F) < g(l_i) - \delta(n)$ and $|F| \leq 2^{l_i}$ for some i and F, then print F. **End of Subroutine**

This subroutine prints all the forbidden sets in some order. Let $F_1, \ldots, F_T$ be that order. Unfortunately we do not know when the subroutine will print the last forbidden set. In other words, we do not know the number T of forbidden sets. To overcome this problem, the algorithm will run the subroutine and every time a new forbidden set F_t is printed, the algorithm will construct *candidate sets* $B_0(t), \ldots, B_N(t) \in \mathcal{A}_n$ satisfying $|B_i(t)| \leq 2^{l_i}$ and $C(B_i(t)) \leq g(l_i) + \delta(n)$ and the following condition:

$$\bigcap_{j=0}^{N} B_j(t) \setminus \bigcup_{j=1}^{t} F_j \neq \emptyset \tag{12}$$

for every $t = 0, \ldots, T$. For $t = T$ the set $\bigcup_{j=1}^{t} F_j$ is the union of all forbidden sets, which guarantees the bounds $g(l_i) - \delta(n) \leq g_x(l_i) \leq g(l_i) + \delta(n)$ for all x in the set in the left-hand side (LHS) of (12). Then we will prove that these bounds imply that $g(l) - \delta(n) \leq g_x(l) \leq g(l) + \delta(n)$ for *every* $l = 0, \ldots, n$. Each time a new forbidden set appears (that is, for every $t = 1, \ldots, T$) we will need to update candidate sets so that (12) remains true. To do that we will maintain a stronger condition than just nonemptiness of the LHS of (12). Namely, we will maintain the following invariant: for every $i = 0, 1, \ldots, N$

$$\left| \bigcap_{j=0}^{i} B_j(t) \setminus \bigcup_{j=1}^{t} F_j \right| \geq 2^{l_i - i - 1} \alpha_n^{-i}. \tag{13}$$

Note that for $i = N$ inequality (13) implies (12).

```
Algorithm $(n, \mathcal{A}_n, g(l_0), g(l_1), \ldots, g(l_N))$:
    Initialize. Recall that $l_0 = n$. Define the set $B_t(0) = \{0,1\}^n$ for every $t$. This set is in $\mathcal{A}_n$ by property 1.
    for $i := 1, \ldots, N$ do
    Assume inductively that $|B_0(0) \bigcap B_1(0) \bigcap \cdots \bigcap B_{i-1}(0)| \geq 2^{l_{i-1}} \alpha_n^{-i+1}$, where $\alpha_n$ denotes a polynomial upper bound of the covering coefficient of distortion family $\mathcal{A}_n$ existing by property 4. (The value $\alpha_n$ can be computed from $n$.) Note that this inequality is satisfied for $i = 1$. Construct $B_i(0)$ by covering $B_{i-1}(0)$ by at most $\alpha_n 2^{l_{i-1} - l_i}$ sets of cardinality at most $2^{l_i}$ (this cover exists in $\mathcal{A}_n$ by property 4). Trivially, this cover also covers $B_0(0) \bigcap \ldots \bigcap B_{i-1}(0)$. The intersection of at least one of the covering sets with $B_0(0) \bigcap \ldots \bigcap B_{i-1}(0)$ has cardinality at least
        $\frac{2^{l_{i-1}} \alpha_n^{-i+1}}{\alpha_n 2^{l_{i-1} - l_i}} = 2^{l_i} \alpha_n^{-i}.$
```

Let $B_i(0)$ by the first such covering set in a given standard order. **od**

Notice that after the Initialization the invariant (13) is true for $t = 0$, as $\bigcup_{j=1}^{t} F_j = \emptyset$. For every $t = 1, 2, \ldots$ perform the following steps 1 and 2 maintaining the invariant (13):

Step 1. Run the subroutine and wait until tth forbidden set F_t is printed (if $t > T$ the algorithms waits forever and never proceeds to Step 2).

Step 2.

Case 1. For every $i = 0, 1, \ldots, N$ we have

$$\left| \bigcap_{j=0}^{i} B_j(t-1) \setminus \bigcup_{j=1}^{t} F_j \right| \geq 2^{l_i - i - 1} \alpha_n^{-i}. \tag{14}$$

Note the this inequality has one more forbidden set compared to the invariant (13) for $t-1$ (the argument in $B_j(t-1)$), and thus may be false. If that is the case, then we let $B_i(t) = B_i(t-1)$ for every $i = 1, \ldots, N$ (this setting maintains invariant (13)).

Case 2. Assume that (14) is false for some index i. In this case find the least such index (we will use later that (14) is true for all $i' < i$).

We claim that $i > 0$. That is, the inequality (14) is true for $i = 0$. In other words, the cardinality of $F_1 \bigcup \cdots \bigcup F_t$ is not larger than half of the cardinality of $B_0(t-1) = \{0,1\}^n$. Indeed, for every fixed i the total cardinality of all the sets of simultaneously cardinality at most 2^{l_i} and Kolmogorov complexity less than $g(l_i) - \delta(n)$ does not exceed $2^{g(l_i)-\delta(n)} 2^{l_i}$. Therefore, the total number of elements in $\bigcup_{j=1}^{t} F_t$ is at most

$$\begin{aligned} &\sum_{i=0}^{N} 2^{g(l_i) - \delta(n) + l_i} \\ &\quad \leq (N+1) 2^{g(n) - \delta(n) + n} \\ &\quad = (N+1) 2^{n - \delta(n)} \ll 2^{n-1} = \frac{1}{2} \left|\{0,1\}^n\right| \end{aligned}$$

where the first inequality follows since the function $g(l) + l$ is monotonic nondecreasing, the first equality since $g(n) = 0$ by definition, and the last inequality since we will set $\delta(n)$ at order of magnitude $\sqrt{n \log n}$.

First let $B_k(t) = B_k(t-1)$ for all $k < i$ (this maintains invariant (13) for all $k < i$). To define $B_i(t)$ find a covering of $B_{i-1}(t)$ by at most $\alpha_n 2^{l_{i-1} - l_i}$ sets in $\mathcal{A}_n$ of cardinality at most 2^{l_i}. Since (14) is true for index $i - 1$, we have

$$\left| \bigcap_{j=0}^{i-1} B_j(t) \setminus \bigcup_{j=1}^{t} F_t \right| \geq 2^{l_{i-1} - i} \alpha_n^{-i+1}. \tag{15}$$

Thus the greatest cardinality of an intersection of the set in (15) with a covering set is at least

$$\frac{2^{l_{i-1} - i} \alpha_n^{-i+1}}{\alpha_n 2^{l_{i-1} - l_i}} = 2^{l_i - i} \alpha_n^{-i}.$$

Let $B_i(t)$ be the first such covering set in standard order. Note that $2^{l_i - i} \alpha_n^{-i}$ is at least twice the threshold required by invariant (13). Use the same procedure to obtain successively $B_{i+1}(t), \ldots, B_N(t)$.

End of Algorithm

Although the algorithm does not halt, at some unknown time the last forbidden set F_T is enumerated. After this time the candidate sets are not changed anymore. The invariant (13) with $i = N$ shows that the cardinality of the set in the LHS of (12) is positive, and hence the set is not empty.

Next we show that $C(B_i(t)) \leq g(l_i) + \delta(n)$ for every i and every $t = 1, \ldots, T$. We will see that to this end it suffices to upperbound the number of changes of each candidate set.

Definition 13: Let m_i be the *number of changes of* B_i defined by $m_i = |\{t : B_i(t) \neq B_i(t-1), 1 \leq t \leq T\}|$ for $0 \leq i \leq N$.

Claim 2: $m_i \leq 2^{g(l_i)+i}$ for $0 \leq i \leq N$.

Proof: The Claim is proved by induction on i. For $i = 0$ the claim is true, since $l_0 = n$ and $g(n) = 0$ while $m_0 = 0$ by initialization in the Algorithm ($B(0)$ never changes).

($i > 0$): assume that the Claim is satisfied for every j with $0 \leq j < i$. We will prove that $m_i \leq 2^{g(l_i)+i}$ by counting separately the number of changes of B_i of different types.

Change of type 1. The set B_i is changed when (14) is false for an index strictly less than i. The number of these changes is at most

$$m_{i-1} \leq 2^{g(l_{i-1})+i-1} \leq 2^{g(l_i)+i-1},$$

where the first inequality follows from the inductive assumption, and the second inequality by the property of g that it is nonincreasing. Namely, since $l_{i-1} > l_i$ we have $g(l_{i-1}) \leq g(l_i)$.

Change of type 2. The inequality (13) is false for i and is true for all smaller indexes.

Change of type 2a. After the last change of B_i at least one j-forbidden set for some $j < i$ has been enumerated. The number of changes of this type is at most the number of j-forbidden sets for $j = 0, \ldots, i-1$. For every such j these forbidden sets have by definition Kolmogorov complexity less than $g(l_j) - \delta(n)$. Since $l_j \geq l_i$ and g is monotonic nonincreasing we have $g(l_j) \leq g(l_i)$. Because there are at most N of these j's, the number of such forbidden sets is at most

$$N 2^{g(l_i) - \delta(n)} \ll 2^{g(l_i)}$$

since we will later choose $\delta(n)$ of order $\sqrt{n \log n}$,

Change of type 2b. Finally, for every change of this type, between the last change of B_i and the current one no candidate sets with indexes less than i have been changed and no j-forbidden sets with $j < i$ have been enumerated. Since after the last change of B_i the cardinality of the set in the LHS of (13) was at least $2^{l_i - i} \alpha_n^{-i}$, which is twice the threshold in the RHS by the restoration of the invariant in the Algorithm Step 2, Case 2, the following must hold. The cardinality of $\bigcup_{j=1}^{t} F_j$ increased by at least $2^{l_i - i - 1} \alpha_n^{-i}$ since the last change of B_i, and this must be due to enumerating j-forbidden sets for $j = i, \ldots, N$. For every such j every j-forbidden set has cardinality at most 2^{l_j} and Kolmogorov complexity less than $g(l_j) - \delta(n)$. Hence, the total number of elements in all j-forbidden sets is less than $2^{l_j} 2^{g(l_j) - \delta(n)}$. Since $j \geq i$ and hence $l_j \leq l_i$ while

$g(l)+l$ is monotonic nondecreasing we have $g(l_j)+l_j \leq g(l_i)+l_i$. Because there are at most $N+1$ of these j's, the total number of elements in all those sets does not exceed $M=(N+1)2^{g(l_i)-\delta(n)+l_i}$. The number of changes of this type is not more than the total number M of elements involved divided by the increments of size $2^{l_i-i-1}\alpha_n^{-i}$. Hence it is not more than

$$(N+1)2^{g(l_i)-\delta(n)}2^{i+1}\alpha_n^i.$$

Let

$$\begin{aligned}\delta(n) &\geq \log\left((N+1)2^{i+10}\alpha_n^i\right) \quad \text{and}\\ \delta(n) &= O(N\log(2\alpha_n))\\ &= O\left(\sqrt{n/\log n}\log(2\alpha_n)\right)\\ &= O\left(\sqrt{n\log n}\right) \qquad (16)\end{aligned}$$

where the last equality uses that α_n is polynomial in n by property 4. Then, the number of changes of type 2b is much less than $2^{g(l_i)}$. The value of $\delta(n)$ can be computed from n.

Summing the numbers of changes of types 1, 2a, and 2b we obtain $m_i \leq 2^{g(l_i)+i}$, completing the induction. ■

Claim 3: Every x in the nonempty set (12) satisfies $|g_x(l_i)-g(l_i)| \leq \delta(n)$ with $\delta(n)=O(\sqrt{n\log n})$ for $i=0,1,\ldots,N$.

Proof: By construction x is not an element of any forbidden set in $\bigcup_{t=1}^T F_t$, and, therefore

$$g_x(l_i) \geq g(l_i)-\delta(n)$$

for every $i=0,1,\ldots,N$. By construction $|B_i(T)| \leq 2^{l_i}$, and to finish the proof it remains to show that $C(B_i(T)) \leq g(l_i)+\delta(n)$ so that $g_x(l_i) \leq g(l_i)+\delta(n)$, for $i=0,1,\ldots,N$. Fix i. The set $B_i(T)$ can be described by a constant length program, that is $O(1)$ bits, that runs the Algorithm and uses the following information:

- A description of i in $\log N \leq \log n$ bits.
- A description of the distortion family $\mathcal{A}_n$ in $O(\log n)$ bits by property 3.
- The values of g in the points $l_0,\ldots,l_N$ in $N\log n = \sqrt{n\log n}$ bits.
- The description of n in $O(\log n)$ bits.
- The total number m_i of changes (Case 2 in the Algorithm) to intermediate versions of B_i in $\log m_i$ bits.

We count the number of bits in the description of $B_i(T)$. The description is effective and by Claim 2 with $i \leq N = \sqrt{n/\log n}$ it takes at most $g(l_i)+O(\sqrt{n\log n})$ bits. So this is an upper bound on the Kolmogorov complexity $C(B_i(T))$. Therefore, for some $\delta(n)$ satisfying (16) we have

$$g_x(l_i) \leq g(l_i)+\delta(n),$$

for every $i=0,1,\ldots,N$. The claim follows from the first and the last displayed equation in the proof. ■

Let us show that the statement of Claim 3 holds not only for the subsequence of values $l_0,l_1,\ldots,l_N$ but for every $l=0,1,\ldots,n$,

Let $l_i \leq l \leq l_{i-1}$. Both functions $g(l), g_x(l)$ are nonincreasing so that

$$\begin{aligned}g(l) &\in [g(l_{i-1}), g(l_i)]\\ g_x(l) &\in [g_x(l_{i-1}), g_x(l_i)]\\ &\subseteq \Big[g(l_{i-1})-O\left(\sqrt{n\log n}\right),\\ &\qquad g(l_i)+O\left(\sqrt{n\log n}\right)\Big].\end{aligned}$$

By the spacing of the sequence of l_i's the length of the segment $[g(l_{i-1}), g(l_i)]$ is at most

$$g(l_i)-g(l_{i-1}) \leq l_{i-1}-l_i = \sqrt{n\log n}.$$

If there is an x such that Claim 3 holds for every l_i with $i=0,\ldots,N$, then it follows from the above that $|g(l)-g_x(l)| \leq \sqrt{n\log n}+O(\sqrt{n\log n})$ for every $l=0,1,\ldots,n$. ■

Proof: of Theorem 2. We start with Lemma 6 stating a combinatorial fact that is interesting in its own right, as explained further in Remark 8.

Lemma 6: Let n,m,k be natural numbers and x a string of length n. Let $\mathcal{B}$ be a family of subsets of $\{0,1\}^n$ and $\mathcal{B}(x)=\{B\in\mathcal{B}: x\in B\}$. If $\mathcal{B}(x)$ has at least 2^m elements (that is, sets) of Kolmogorov complexity less than k, then there is an element in $\mathcal{B}(x)$ of Kolmogorov complexity at most $k-m+O(C(\mathcal{B})+\log n+\log k+\log m)$.

Proof: Consider a game between Alice and Bob. They alternate moves starting with Alice's move. A move of Alice consists in producing a subset of $\{0,1\}^n$. A move of Bob consists in marking some sets previously produced by Alice (the number of marked sets can be 0). Bob wins if after every one of his moves every $x\in\{0,1\}^n$ that is covered by at least 2^m of Alice's sets belongs to a marked set. The length of a play is decided by Alice. She may stop the game after any of Bob's moves. However, the total number of her moves (and hence Bob's moves) must be less than 2^k. (It is easy to see that without loss of generality we may assume that Alice makes exactly 2^k-1 moves.) Bob can easily win if he marks every set produced by Alice. However, we want to minimize the total number of marked sets.

Claim 4: Bob has a winning strategy that marks at most $O(2^{k-m}k^2n)$ sets.

Proof: We present an explicit strategy for Bob, which consists in in executing at every move $t=1,2,\ldots,2^k-1$ the following algorithm for the sequence $A_1,A_2,\ldots,A_t$ which has been produced by Alice until then.

- **Step 1**. Let 2^j be the largest power of 2 dividing t. Consider the last 2^j sets in the sequence $A_1,A_2,\ldots,A_t$ and call them $D_1,\ldots,D_{2^j}$.
- **Step 2**. Let T be the set of x's that occur in at least $2^m/k$ of the sets $D_1,\ldots,D_{2^j}$. Let D_p be a set such that $|D_p\bigcap T|$ is maximal. Mark D_p (if there is more than one then choose the one with p least) and remove all elements of $D_p\bigcap T$ from T. Call the resulting set T_1. Let D_q be a set such that $|D_q\bigcap T_1|$ is maximal (if there is more than one then choose the one with q least). After removing all elements of $D_q\bigcap T_1$ from T_1 we obtain a set T_2. Repeat the argument until we obtain $T_{e_j}=\emptyset$.

First, for the j above we have $e_j \leq \lceil 2^{j-m} kn \ln 2 \rceil$. This is proved as follows. We have

$$\sum_{i=1}^{2^j} \left| D_i \bigcap T \right| \geq |T| 2^m / k$$

since every $x \in T$ is counted at least $2^m/k$ times in the sum in the LHS. Thus, there is a set in the list $D_1, \ldots, D_{2^j}$ such that the cardinality of its intersection with T is at least 2^{-j} times the RHS. By the choice of D_p it is such a set and we have $|D_p \bigcap T| \geq |T| 2^{m-j}/k$.

The set T has lost at least a $(2^{m-j}/k)$th fraction of its elements, that is, $|T_1| \leq |T|(1 - 2^{m-j}/k)$. Since $T_1 \subseteq T$, obviously every element of T_1 (still) occurs in at least $2^m/k$ of the sets $D_1, \ldots, D_{2^j}$. Thus we can repeat the argument and mark a set D_q with $|D_q \bigcap T_1| \geq |T_1| 2^{m-j}/k$. After removing all elements of $D_q \bigcap T_1$ from T_1 we obtain a set T_2 that is at most a $(1 - 2^{m-j}/k)$th fraction of T_1, that is, $|T_2| \leq |T_1|(1 - 2^{m-j}/k)$.

Recall that we repeat the procedure e_j times where e_j is the number of repetitions until $T_{e_j} = \emptyset$. It follows that $e_j \leq \lceil 2^{j-m} kn \ln 2 \rceil$, since

$$|T|(1 - 2^{m-j}/k)^{2^{j-m} kn \ln 2} < |T| e^{-n \ln 2} = |T| 2^{-n} \leq 1.$$

Second, for every fixed $j = 0, 1, \ldots, k-1$ there are at most 2^{k-j} different t's ($t = 1, 2, \ldots, 2^k - 1$) divisible by 2^j and the number $d_j = 2^{k-j} e_j$ of marked sets we need to use for this j satisfies $d_j \leq 2^{k-j} 2^{j-m} kn \ln 2 = 2^{k-m} kn \ln 2$. For all $j = 0, \ldots, k-1$ together we use a total number of marked sets of at most

$$\sum_{j=0}^{k-1} d_j \leq 2^{k-m} k^2 n \ln 2.$$

In this way, after every move $t = 1, 2, \ldots, 2^k - 1$ of Bob, every x occurring in 2^m of Alice's sets belongs to a marked set of Bob. This can be seen as follows. Assume to the contrary, that there is an x that occurs in 2^m of Alice's sets following move t of Bob, and x belongs to no set marked by Bob in step t or earlier. Let $t = 2^{j_1} + 2^{j_2} + \cdots$ with $j_1 > j_2 > \cdots$ be the binary expansion of t. By Bob's strategy, the element x occurs less than $2^m/k$ times in the first segment of 2^{j_1} sets of Alice, less than $2^m/k$ times in the next segment of 2^{j_2} of Alice's sets, and so on. Thus its total number of occurrences among the t first sets of Alice is strictly less than $k 2^m / k = 2^m$. ■

Let us finish the proof of the Lemma 6. Given the list of $\mathcal{B}$, recursively enumerate the sets in $\mathcal{B}$ of Kolmogorov complexity less than k, say $B_1, B_2, \ldots, B_T$ with $T < 2^k$, and consider this list as a particular sequence of moves by Alice. Use Bob's strategy of Claim 4 against Alice's sequence as above. Note that recursive enumeration of the sets in $\mathcal{B}$ of Kolmogorov complexity less than k means that eventually all such sets will be produced, although we do not know when the last one is produced. This only means that the time between moves is unknown, but the alternating moves between Alice and Bob are deterministic and sequential. According to Claim 4, Bob's strategy marks at most $O(2^{k-m} k^2 n)$ sets. These marked sets cover every string occurring at least in 2^m of the sets $B_1, B_2, \ldots, B_T$. We do not know when the last set B_T appears in this list, but Bob's winning strategy of Claim 4 ensures that immediately after recursively enumerating B_t ($t \leq T$) in the list every string that occurs in 2^m sets in the initial segment $B_1, B_2, \ldots B_t$ is covered by a marked set. The Kolmogorov complexity $C(B_i)$ of every marked set B_i in the list $B_1, B_2, \ldots, B_T$ is upper bounded by the logarithm of the number of marked sets, that is $k - m + O(\log k + \log n)$, plus the description of $\mathcal{B}, k, m$, and n including separators in $O(C(\mathcal{B}) + \log k + \log m + \log n)$ bits. ■

We continue the proof of the theorem. Let the distortion family $\mathcal{A}$ satisfy properties 2 and 3. Consider the subfamily $\mathcal{B}$ of $\mathcal{A}_n$ consisting of all sets A with $\lceil \log A \rceil = \lceil \log B \rceil$. Let $\mathcal{B}(x)$ be the family $\{B \in \mathcal{B} : x \in B\}$ and N the number of sets in $\mathcal{B}(x)$ of Kolmogorov complexity at most $C(B)$.

Given $x, \lceil \log B \rceil, \mathcal{A}_n$ and $C(B)$ we can generate all $A \in \mathcal{B}(x)$ of Kolmogorov complexity at most $C(B)$. Then we can describe B by its index among the generated sets. This shows that the description length $C(B \,|\, x) \leq \log N$ (ignoring an additive term of order $O(\log C(B) + \log n)$ which suffices since $C(\lceil \log B \rceil)$ and $C(\mathcal{A}_n)$ are both $O(\log n)$).

Since $C(\mathcal{A}_n) = O(\log n)$ by property 3, $\mathcal{B} \subseteq \mathcal{A}_n$ while every set $A \in \mathcal{B}$ satisfies $\lceil \log |A| \rceil = \lceil \log |B| \rceil \leq n$, we have $C(\mathcal{B}) = O(\log n)$. Let $k = C(B) + 1$ and $m = \lfloor \log N \rfloor$, and ignore additive terms of order $O(\log k + \log m + \log n)$. Applying Lemma 6 shows that there is a set $A \in \mathcal{B}(x)$ with $C(A) \leq k - m \leq C(B) - C(B \,|\, x) = I(x : B)$ and therefore proves Theorem 2. ■

Remark 8: Previously an analog of Lemma 6 was known in the case when $\mathcal{B}$ is the class of *all* subsets $\{0,1\}^n$ of *fixed* cardinality 2^l. For $l = 0$ this is in [9, Exercise 4.3.8 (second edition) and 4.3.9 (third edition)]: If a string x has at least 2^m descriptions of length at most k (p is called a description of x if $U(p) = x$ where U is the reference Turing machine), then $C(x) \leq k - m + O(\log k + \log m)$. Reference [22] generalizes this to all $l > 0$: If a string belongs to at least 2^m sets B of cardinality 2^l and Kolmogorov complexity $C(B) \leq k$, then x belongs to a set A of cardinality 2^l and Kolmogorov complexity $C(A) \leq k - m + O(\log m + \log k + \log l)$. ◊

Remark 9: Probabilistic proof of Claim 4. Consider a new game that has the same rules and one additional rule: Bob looses if he marks more than $2^{k-m+1}(n+1) \ln 2$ sets. We will prove that in this game Bob has a winning strategy.

Assume the contrary: Bob has no winning strategy. Since the number of moves in the game is finite (less than 2^k), this implies that Alice has a winning strategy.

Fix a winning strategy S of Alice. To obtain a contradiction we design a randomized strategy for Bob that beats Alice's strategy S with positive probability. Bob's strategy is very simple: mark every set produced by Alice with probability $p = 2^{-m}(n+1) \ln 2$.

Claim 5: (i) With probability more than $\frac{1}{2}$, following every move of Bob every element occurring in at least 2^m of Alice's sets is covered by a marked set of Bob.

(ii) With probability more than $\frac{1}{2}$, Bob marks at most $2^{k-m+1}(n+1) \ln 2$ sets.

Proof: (i) Fix x and estimate the probability that there is move of Bob following which x belongs to 2^m of Alice's sets but belongs to no marked set of Bob.

Let R_i be the event "following a move of Bob, string x occurs at least in i sets of Alice but none of them is marked." Let us prove by induction that

$$\Pr[R_i] \le (1-p)^i.$$

For $i = 0$ the statement is trivial. To prove the induction step we need to show that $\Pr[R_{i+1}|R_i] \le 1-p$.

Let $z = z_1, z_2, \ldots, z_t$ be a sequence of decisions by Bob: $z_j = 1$ if Bob marks the jth set produced by Alice and $z_j = 0$ otherwise. Call z *bad* if following Bob's tth move it happens for the first time that x belongs to i sets produced by Alice by move t but none of them is marked. Then R_i is the disjoint union of the events "Bob has made the decisions z" (denoted by Q_z) over all bad z. Thus it is enough to prove that

$$\Pr[R_{i+1} \,|\, Q_z] \le 1-p.$$

Given that Bob has made the decisions z, the event R_{i+1} means that after those decisions the strategy S will at some time in the future produce the $(i+1)$st set with member x but Bob will not mark it. Bob's decision not to mark that set does not depend on any previous decision and is made with probability $1-p$. Hence

$$\begin{aligned}\Pr[R_{i+1} \,|\, Q_z] &\\ = \Pr[&\text{Alice produces the } (i+1)\text{st set with member} \\ &x \,|\, Q_z] \cdot (1-p) \le 1-p.\end{aligned}$$

The induction step is proved. Therefore, $\Pr[R_{2^m}] \le (1-p)^{2^m} < e^{-p2^m} = 2^{-n-1}$, where the last equality follows by choice of p.

(ii) The expected number of marked sets is $p2^k$. Thus the probability that it exceeds $p2^{k+1}$ is less than $\frac{1}{2}$. ■

It follows from Claim 5 that there exists a strategy by Bob that marks at most $2^{k-m+1}(n+1)\ln 2$ sets out of Alice's produced 2^k sets, and following every move of Bob every element occurring in at least 2^m of Alice's sets is covered by a marked set of Bob. Note that we have proved that this strategy of Bob exists but we have not constructed it. Given n, k and m, the number of games is finite, and a winning strategy for Bob can be found by brute force search. ■

Proof: of Theorem 3. Let $B \subseteq \{0,1\}^n$ be a set containing string x. Define the *sufficiency deficiency of x in B* by

$$\log|B| + C(B) - C(x).$$

This is the number of extra bits incurred by the two-part code for x using B compared to the most optimal one-part code of x using $C(x)$ bits. We relate this quantity with the randomness deficiency $\delta(x \,|\, B) = \log|B| - C(x \,|\, B)$ of x in the set B. The randomness deficiency is always less than the sufficiency deficiency, and the difference between them is equal to $C(B \,|\, x)$:

$$\log|B| + C(B) - C(x) - \delta(x \,|\, B) = C(B \,|\, x), \qquad (17)$$

where the equality follows from the symmetry of information (10), ignoring here and later in the proof additive terms of order $O(\log C(B) + \log n)$.

By Theorem 2, which assumes that properties 2 and 3 hold for the distortion family $\mathcal{A}$, there is $A \in \mathcal{A}(x)$ with $\lceil \log|A| \rceil = \lceil \log|B| \rceil$ and $C(A) \le C(B) - C(B \,|\, x)$. Since A_x is a set of minimal Kolmogorov complexity among such A we have $C(A_x) \le C(B) - C(B \,|\, x)$. Therefore

$$\begin{aligned}&C(A_x) + \log|A_x| - C(x)\\ &\quad\le C(B) - C(B \,|\, x) + \log|A_x| - C(x)\\ &\quad= C(B) - C(B \,|\, x) + \log|B| - C(x)\\ &\quad= \delta(x \,|\, B),\end{aligned}$$

where the last equality is true by (17). ■

ACKNOWLEDGMENT

The authors would like to thank A. K. Shen for helpful suggestions. A. A. Muchnik gave the probabilistic proof of Claim 4 in Remark 9 after having seen the deterministic proof. Such a probabilistic proof was independently proposed by M. Koucký. The authors would also like to thank the referees for their constructive comments; one referee pointed out that yet another example would be the case of Euclidean balls with the usual Euclidean distance, where the important Property 4 is proved in for example [23].

REFERENCES

[1] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] T. Berger and J. D. Gibson, "Lossy source coding," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2693–2723, 1998.

[3] M. Burrows and D. J. Wheeler, A Block-Sorting Lossless Data Compression Algorithm. Digital Equip. Corp., Syst. Res. Center, Tech. Rep. 124, May 1994.

[4] S. C. Chang, B. Yu, and M. Vetterli, "Image denoising via lossy compression and wavelet thresholding," in *Proc. Int. Conf. Image Process. (ICIP'97)*, 1997, vol. 1, pp. 604–607.

[5] D. Donoho, "The Kolmogorov sampler," *Ann. Statist.*, submitted for publication.

[6] P. Gács, J. Tromp, and P. M. B. Vitányi, "Algorithmic statistics," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2443–2463, 2001.

[7] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Problems Inf. Transmiss.*, vol. 1, no. 1, pp. 1–7, 1965.

[8] A. N. Kolmogorov, "Complexity of algorithms and objective definition of randomness," a talk at Moscow Math. Soc. meeting 4/16/1974.," *Uspekhi Mat. Nauk*, vol. 29, no. 4, p. 155, 1974, An abstract available in English translation in [22].

[9] M. Li and P. M. B. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd, 3rd ed. New York: Springer-Verlag, 1997, 2008.

[10] M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.

[11] M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitanyi, "The similarity metric," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3250–3264, 2004.

[12] B. K. Natarajan, "Filtering random noise from deterministic signals via data compression," *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2595–2605, 1995.

[13] J. Muramatsu and F. Kanaya, "Distortion-complexity and rate-distortion function," *IEICE Trans. Fund.*, vol. E77-A, no. 8, pp. 1224–1229, 1994.

[14] A. Rumyantsev, "Transmission of information through a noisy channel in Kolmogorov complexity setting," *Vestnik MGU, Seriya Matematika i Mechanika* (Russian), to appear.

[15] S. de Rooij and P. M. B. Vitanyi, "Approximating rate-distortion graphs of individual data: Experiments in lossy compression and denoising," *IEEE Trans. Comput.*, Submitted. Also: Arxiv preprint cs.IT/0609121, 2006.

[16] N. Saito, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion," in *Wavelets in Geophys.*, E. Foufoula-Georgiou and P. Kumar, Eds. New York: Academic, 1994, pp. 299–324.

[17] C. E. Shannon, "The mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 379–423, pp. 623–656, 1948.

[18] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Convent. Rec., Part 4*, pp. 142–163, 1959.

[19] A. Kh. Shen, "The concept of (α, β)-stochasticity in the Kolmogorov sense, and its properties," *Soviet Math. Dokl.*, vol. 28, no. 1, pp. 295–299, 1983.

[20] D. M. Sow and A. Eleftheriadis, "Complexity distortion theory," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 604–608, 2003.

[21] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, 42:2(1936), 230–265, "Correction", *Ibid.*, 43(1937), 544–546.

[22] N. K. Vereshchagin and P. M. B. Vitányi, "Kolmogorov's structure functions and model selection," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3265–3290, 2004.

[23] J. L. Verger-Gaugry, "Covering a ball with smaller equal balls in R^n," *Discrete and Computational Geometry*, vol. 33, pp. 143–155, 2005.

[24] V. V. V'yugin, "On the defect of randomness of a finite object with respect to measures with given complexity bounds," *SIAM Theory Probab. Appl.*, vol. 32, no. 3, pp. 508–512, 1987.

[25] E.-H. Yang and S.-Y. Shen, "Distortion program-size complexity with respect to a fidelity criterion and rate-distortion function," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 288–292, 1993.

[26] J. Ziv, "Distortion-rate theory for individual sequences," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 2, pp. 137–143, 1980.

Nikolai K. Vereshchagin received the Ph.D. degree from the Moscow State University, Moscow, Russia, in 1986.

He is a Professor of Mathematical Logic and Theory of Algorithms, Moscow State University, Russia. He has worked on algorithms in number theory, decidable theories, computational complexity, and Kolmogorov complexity. Together with A. Shen, he coauthored *Computable Functions* and *Basic Set Theory* (Moscow: MCNMO, 1999), in Russian, both of which have been published by the American Mathematical Society in English translation. Web page: http://lpcs.math.msu.su/ver/

Paul M. B. Vitányi received the Ph.D. degree from the Free University, Amsterdam, The Netherlands, in 1978.

He is a CWI Fellow with the National Research Institute for Mathematics and Computer Science, Netherlands, CWI, and Professor of Computer Science, University of Amsterdam, The Netherlands. He has worked on cellular automata, computational complexity, distributed and parallel computing, machine learning and prediction, physics of computation, Kolmogorov complexity, and information theory. He has published approximately 200 research papers and some books. Together with M. Li, they pioneered applications of Kolmogorov complexity and coauthored *An Introduction to Kolmogorov Complexity and its Applications* (Springer-Verlag, 1993) (2nd ed., 1997, 3rd ed., 2008), parts of which have been translated into Chinese, Russian, and Japanese. Web page: http://www.cwi.nl/paulv/.

Prof. Vitanyi received a Knighthood in 2007. He serves on the editorial boards of *Distributed Computing* (1987–2003), *Information Processing Letters*, *Theory of Computing Systems*, *Parallel Processing Letters*, *International Journal of Foundations of Computer Science*, *Entropy*, *Information*, *Journal of Computer and Systems Sciences* (guest editor), and elsewhere.